2016

Graphical User Interface for Antimicrobial Peptide Database

Tripti Garg
University of Rhode Island, tripti_garg@my.uri.edu

GRAPHICAL USER INTERFACE FOR ANTIMICROBIAL PEPTIDE DATABASE

BY

TRIPTI GARG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

COMPUTER SCIENCE

UNIVERSITY OF RHODE ISLAND

2016

MASTER OF SCIENCE THESIS IN COMPUTER SCIENCE

OF

TRIPTI GARG

APPROVED:

Thesis Committee:

Major Professor Joan Peckham

Lenore M. Martin

Lisa DiPippo

Ying Zhang

Nasser H. Zawia
DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND
2016

ABSTRACT

This thesis describes the design and implementation of Anti-Microbial Peptide Editable Database ("AMPed"), a tool to enable researchers to efficiently search, display, manipulate and store data about antimicrobial peptides. The tool is implemented as a secure website, created primarily using PHP and MySQL. The website exposes the data collected from a wide variety of sources via a set of common rules and nomenclature making it easy and intuitive for researchers to work with it. The website was evaluated on a variety of web browsers, as well as with a variety of users. It solves the need of non-technical users who spend hours searching and correlating relevant peptide data from a variety of sources for their research needs. This thesis primarily focused on creating and normalizing the database, creating modular search queries and building the web interface. This thesis also introduces a new way to document components of complex and interactive webpages.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my major professors, Dr. Joan Peckham and Dr. Lenore M. Martin, for giving me the opportunity to be part of the AMPed project. They were an inspiration to me throughout the entire Masters Program. I greatly appreciate their valuable guidance, generously lending me their time and sharing with me their expertise. It is primarily because of their continued support that I was able to complete this thesis.

I would also like to thank all the committee members of this thesis Dr. Lisa DiPippo and Dr. Ying Zhang, Department Chair Dr. Gerard M. Baudet and Dr. Jean-Yves Hervé for their ideas, advice and criticism.

A sincere appreciation is extended to George Konstantinidis, who initiated the AMPed database project.

My sincere appreciation is extended to the staff of the Computer Science department Kevin Bryan for his help and guidance on database connectivity and Lorraine Berube for guiding and helping me on thesis submission.

I would also like to extend my deepest gratitude to my husband, Chanchal Gupta, as without his encouragement I would not have had a chance to be at URI. Also special thanks to my family and friends for their encouragement and admiration.

TABLE OF CONTENTS

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

CHAPTER 1

INTRODUCTION: Project Description, Goals, Motivation and Strategies

The Web has become an increasingly large part of our culture as its availability has increased in the past two decades. Its user base has expanded from the original tech- savvy core group of people to a wide range of people with fewer or no technical skills that read, search and/or share their own content.

Currently, the Anti-Microbial Peptide Editable Database ("AMPed") is being developed at URI for research on antimicrobial proteins up to 100 amino acids in length. AMPed is an annotated collection of antimicrobial peptides that are sourced from online repositories, journals, and the existing large publicly available databases. It is today very difficult for a non-technical user to access and work with the data in the AMPed database given its variety and complexity. Every day many users (Dr. Lenore Martin's research team) get frustrated when trying to use AMPed directly. AMPed has high quality data, but when the users are not able to effectively use it for their tasks and needs, it becomes almost irrelevant to them. In fact, a simple search query to pull out data can take hours, and at times days/weeks. So users try to avoid using it directly if they can or try to find technical experts to help them to access the data, utilizing precious time and resources.

This thesis, using agile iterative design and development techniques, Gestalt principles of human interface designs and newest web development practices, builds a normalized database, efficient SQL queries, web interaction diagram and a secure

graphical user interface (GUI) for researchers to efficiently search, display, manipulate and store data about antimicrobial peptides. The Figure1 below shows an overall view of AMPed system.

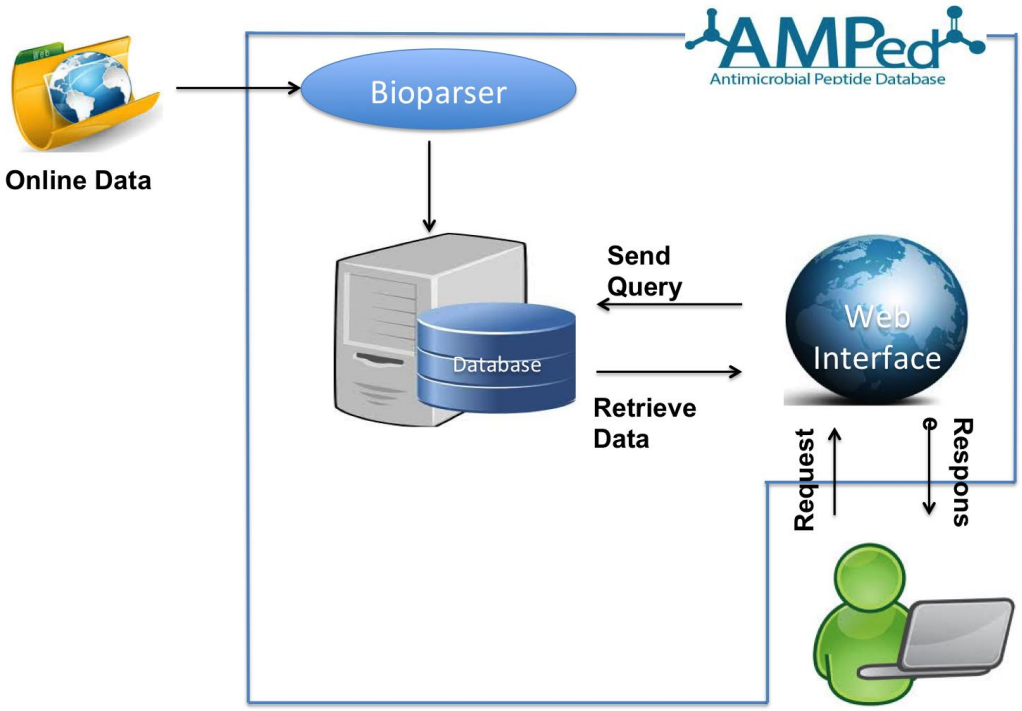


Figure 1: The Overall view of AMPed system

The GUI exposes the data in AMPed collected from a wide variety of sources via a set of common rules and nomenclature. This work significantly enhances the data access, improves the speed & quality of searching, and eliminates the need to manually correlate information gathered from multiple sources, thereby greatly reducing the time researchers have to spend to locate the right data set.

1.1. Review of Literature

1.1.1. Database Normalization

Database normalization is the process of organizing the columns (attributes) and tables (relations) of a relational database to minimize data redundancy and prevent data anomalies. Edgar F. Codd, the inventor of the relational model (RM), introduced the concept of normalization and what we now know as the First normal form (1NF) in 1970. Codd went on to define the Second normal form (2NF) and Third normal form (3NF) in 1971, and Codd and Raymond F. Boyce defined the Boyce-Codd Normal Form (BCNF) in 1974. Informally, a relational database table is often described as "normalized" if it meets Third Normal Form. [1] [2]

AMPed database, developed as part of this thesis and described in detail later, meets the 3NF i.e. its tables are free of insertion, update, and deletion anomalies.

1.1.2. Web Interface Design

There are empirical studies that have identified basic psychological factors that should be considered when designing a good GUI. To design the AMPED GUI, this thesis used the three primary contributing human factors:

- Physical limits of visual acuity
- Limits of absolute memory
- Gestalt Principle

1.1.2.1. Visual Acuity

Visual acuity refers to the ability of the human eye to resolve detail. Studies done by Martin Helander and B.J. Jansen state that "a distance greater than 2.5 degrees from the point of fixation decreases visual acuity by half. Therefore, a circle of radius 2.5 degrees around the point of fixation is what the user can see clearly. At a normal viewing distance of 19 inches, 5 degrees translates into about 1.7 inches. Assuming a standard screen format, 1.7 inches is an area about 14 characters wide and about 7 lines high." [3] [4]

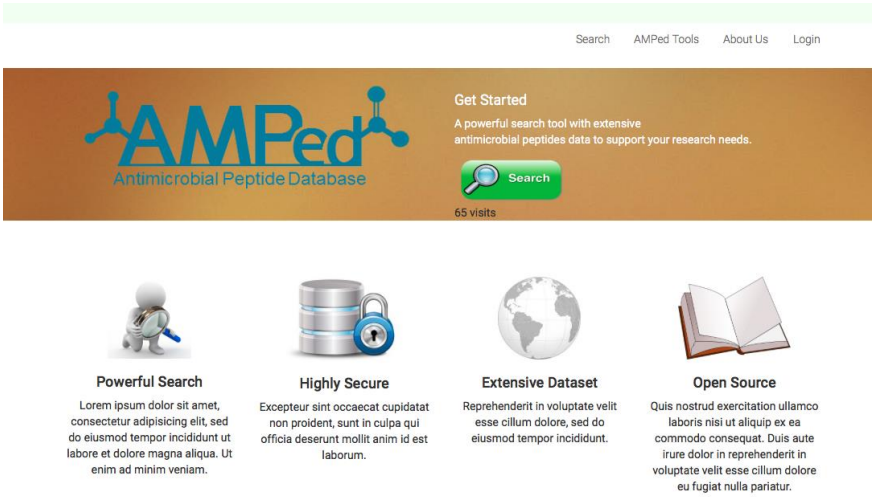


Figure 2: Responsive AMPed Home Page at Big Screen Size (e.g. Desktop)

In Figure 2, the responsive home page of the AMPed is shown at large screen. The same page can be fit to a smaller screen without disturbing the principal of Visual Acuity as shown in Figure 3.

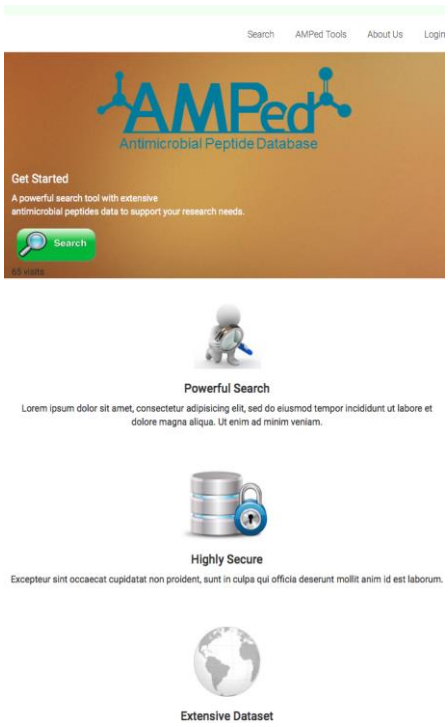


Figure 3: Responsive AMPed Home Page at Small Screen Size (e.g. Smartphone)

AMPed GUI (designed as part of this thesis work) limited the size of icons, menus, dialog boxes etc. to ensure they fit into the limited amount of information the human eye can take at any one time as can been seen in Figure 2 and 3.

This AMPed design also groups information to maintain user focus on one section of the screen. This ensures that the user does not have to constantly move his eyes across the screen causing tiring of the eye due to unnecessary movements.

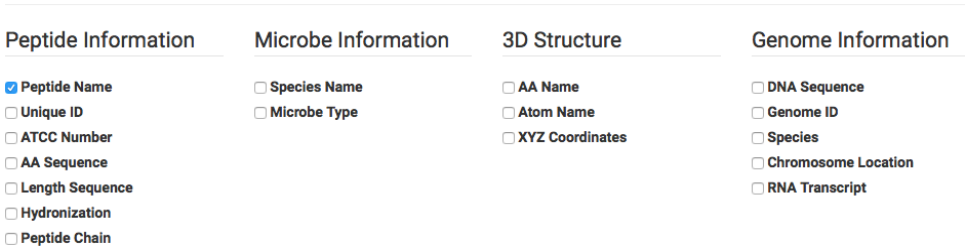


Figure 4: AMPed Search Criteria Page

1.1.2.2. Information Limits

Information limits refer to the amount of data that a person can process at any one time after he/she has fixed a focus point on the user interface. A study done by Miller showed that "absolute identification using one-dimensional criteria was about seven items, plus or minus two. He showed that this limitation also held for memory span." [5] [6] Miller also pointed out "by expanding the identification criteria from one to more dimensions, people could handle more choices and remember more." [5]

AMPed GUI has chunked the information presented to the user (e.g. search criteria) into logical groups. This will ensure that screen is not crowded with data. It also optimizes the number of menu options (e.g. number of search criteria as shown in Figure 4) on the page to develop clean and simple intuitive designs.

1.1.2.3. Gestalt Principle

Gestalt principle or gestalt laws are rules of the organization of perceptual scenes. They were introduced in a seminal paper by Wertheimer (1923/1938), and were further developed by Köhler (1929), Koffka (1935), and Metzger (1936/2006; see review by Todorović, 2007). It describes how people organize visual elements into

groups or unified wholes. Gestalt is also known as the "Law of Simplicity". Per the Gestalt principles, the proper grouping results in a necessary redundancy in selection of information that aids the user. [7] [3]

This project has applied the Gestalt principles to AMPed GUI design and brings together its various elements in a connected, coherent and unified way. For example, it has leveraged the principle of ‘continuation’ via organizing the data in a top-down approach. The summary results page of AMPed, in figure below, shows how the data is grouped and organized top-down.

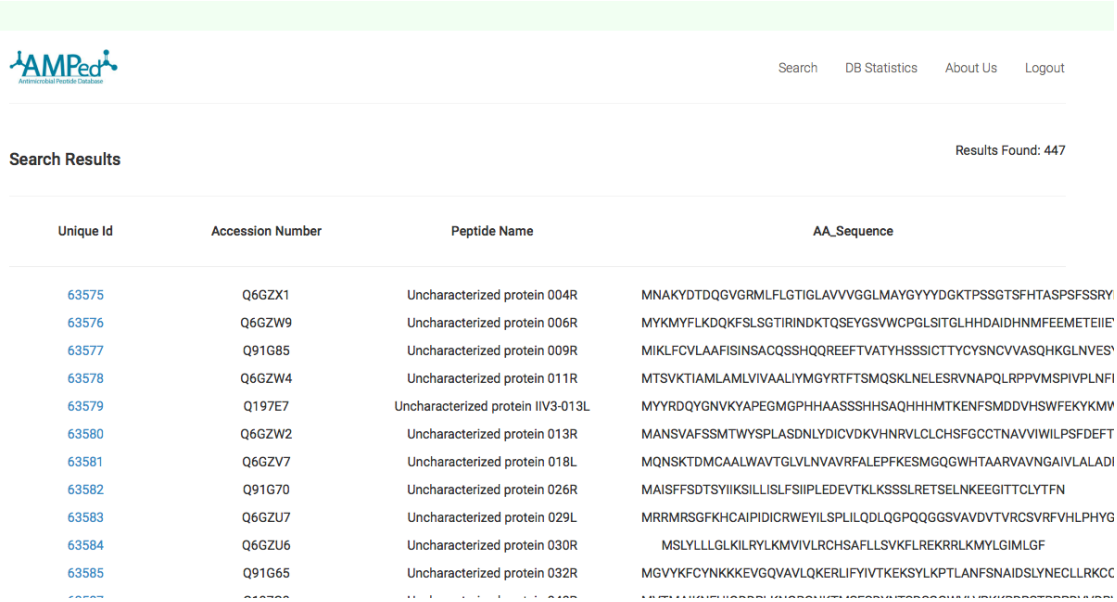


Figure 5: AMPed Summary Search Results Page

It has deployed the principle of ‘similarity’ using a template driven approach that helps the user to easily find similar items. For example, AMPed top header has AMPed Logo with home link and a menu bar that consistently has log-in, log-out,

Search, DB Statistics and About Us links to help user easily navigate through the AMPed website.

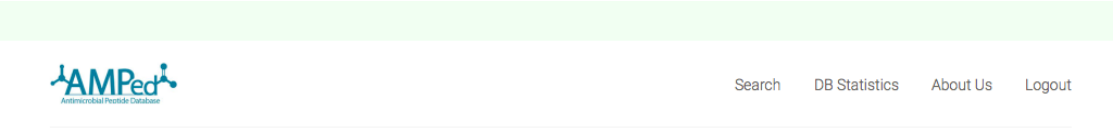


Figure 6: AMPed Header

It uses the principle of 'proximity' to place similar search criteria close together so they are perceived as a logical group. Refer to Figure 6 for example.

1.1.3. Review of existing online Database Repositories

Some repositories of antimicrobial peptides are available online like APD3, YADAMP, LAMP [8][9][10]. We evaluated them all and categorized their performance and currently available features according to the following criteria:

1. Security – Secure user access and privacy
2. Design & Accessibility – user experience and ease of use
3. Website Features – Features and characteristics of web design
4. Website Data – Information provides about the peptide
5. Search – Search capability for peptides and variety of criterions

1. Security: The most important security features for a database of this type begins with assuring users that the database hosts are taking adequate steps to maintain user privacy. Many people nowadays worry about identity theft and exposure to hackers and might be reluctant to use a database they perceive to be insecure. Most of the databases surveyed did not inform the user explicitly about how the information collected about the users will be used. It is also desirable to monitor who is using the

database. Security also includes preventing robotic attacks using some strategy that identifies real human users and defeats automated log-ins. Most of the databases lacked these security measures. AMPed, on the other hand, provides the additional security and protects its users privacy.

2. Design & Accessibility: As described before, web interface design is a key area where the Web site designer must carefully consider the following issues: physical limits of visual acuity, limits of absolute memory, and the Gestalt principle [page3-8]. These basic design principles guided the layout of our graphical user interface (GUI) for AMPed website. Fundamentally, when a new user accesses the database portal for the first time, they should be able to quickly view the database contents in a manner that helps them determine whether the information they are seeking is available, and how easily it can be located. Accordingly, accessibility and design were the first criteria employed when assessing our colleagues existing GUIs. For example, after online search APD3, one arrives at a web page that extends out of the user's field of view in the browser window, which immediately causes the user to have to scroll to see all the information on the home page. The main page does, however, helpfully inform the user "APD contains 2684 antimicrobial peptides from six kingdoms (266 bacteriocins), but it seems to be a statement that is hard-coded into the webpage, and therefore does not allow for increasing the number of peptides in the database without a redesign/update of the page. Additionally, the pages have a lot of information that is not grouped together. This makes APD3 site overly complex to use and difficult to find the relevant information

In LAMP, after the database search, the result page focuses more on comment instead of result of search criterion and general information. If we click on highlighted id number on general information, it moves to comment where users would have expected some more information on it.

In YADAMP search page, the buttons such as search button is labeled "Query YADAMP!" which is confusing and not very intuitive. Further the size of search and reset buttons are very small – easy to miss leading to a not very user-friendly design.

3. Website Data: The main object of all the existing online repositories is to provide information about the antimicrobial peptides. They however, lack consistency and often have multiple versions of the same data making it difficult for the users to get relevant information quickly. Users also have to jump from one site to another or in-between several journals spending their valuable time just to understand the data presented to them. Our main focus with AMPed is to provide all the valuable information about the antimicrobial peptides at one place. Users don't have to jump from one site to another or in-between several journals' as the data is all clearly annotated. For example, as can be seen from the table below, most of the online repositories lack data on 3D structures, amino acids and microbes. AMPed on the other hand provides extensive information on 3D structures, amino acids and microbes alongside the other data, thus making it easier for users and not requiring them to jump in-between sites or journals.

4. Search Feature: Antimicrobial peptide data is very large in volume and is increasing at a rapid pace. This makes search feature a critical component driving use and adoption of the online repositories. While every site provides varying levels of criterion to search through the data, most sites just provide a basic search wherein a user can enter the sequence and website returns all possible matches. YADAMP is the only one that provides many options and logical operators for custom search. However, YADAMP does not provide a description of the different data elements provided on its search page making it more complex for first time users. AMPed also has a robust search feature with many criterions including an option to define a start/end pattern or a pattern anywhere in the sequence, but it is all built in a user friendly way addressing the need of first time or repeat users alike.

5. Website Features: Online repositories like AMPed need to have many functions and can be used in various fashions given the wide set of it's audiences. For example, an individual student can use it for their reference or a professor can use it for advanced research and data storage. So some crucial points for the online presence were studied such as persistent navigation, consistent footer, responsive access, and contact us.

Most of the online repositories compared were not easy to navigate – meaning it was not clear where the menu is and how to get to different pages. For example, on ADP3 website, many pages have no navigation option. Further on detailed information for the search result page, the only option is close window which closes whole website.

Responsive website design enables the website layout to adapt to the screen on

which it's being browsed thus optimizing the use on a tablet, smartphone or desktop. Layouts adjust and images scale to make for a better web experience on these myriad devices. Among all the repositories compared here, AMPed website is the only one which is responsive and can be optimally accessed through tablet and smartphone devices.

If users have questions, they should be easily able to know who to contact and how. Most of the sites compared lacked full contact details. For example, ADP3 and YADAMP did not have any information about the team on their website and LAMP and YADMP provided email as the only contact method.

Category	**YADAMP**	**APD3**	**LAMP**	**AMPed**
Security				
Privacy	N/A	N/A	Store private information in cookie that is accessible	Cookie created but destroyed after the session
Password	No	No	No	Yes
CAPTCHA	No	No	No	Yes
Access Request	No	No	No	Yes
Design and Accessibility				
Physical limits of visual acuity	No	No	Yes	Yes
Limits of absolute memory	No	No	No	Yes
Gestalt principle	Yes	No	Yes	Yes
Website Features				
Responsive	No	No	No	Yes
Persistent Navigation	Yes	No	Yes	Yes
Consistent Footer	Yes	No	Yes	Yes
SiteMap	No	No	No	Yes
Visible Branding	Yes	Yes, but not on every page	Yes	Yes
About the Team	No	No	Yes	Yes

Contact Us	Yes, e-mail only	Yes	Yes, e-mail only	Yes
Number of Visits	No	Yes	No	Yes
Database Statistics	Yes	Yes	Yes	Yes
Website Data				
Peptides	Yes	Yes	Yes	Yes
Genomes	No	No	No	Yes
3D Structures	Yes	Yes	No	Yes
Amino Addresses	No	No	No	Yes
Microbes	Yes	No	No	Yes
Test/Method	Yes	No	No	Yes
Atomic Coordinates	Yes	No	Yes	Yes
Source	Yes	Yes	Yes	Yes
MIC	Yes	Yes	Yes	Yes
Fight Against	Yes	Yes	Yes	Yes
Uniport ID	Yes	No	Yes	Yes
Author	Yes	Yes	Yes	Yes
Length	Yes	Yes	Yes	Yes
Search				
Flexibility	High	High	Low	Medium
Visual Acuity	Low (search buttons are very small)	Low	Low	High
Number of Search Criterion	26	14	10	6
Summary Results	Yes	Yes	Yes	Yes
Detailed Results	Yes	Yes	Yes	Yes
Number of Results Returned	Yes	Yes	No	Yes
Search Criteria for Peptide Sequence	Yes	Yes	No	Yes, Partial and Full Search

Table1: Comparisons of existing online database repositories

Conclusions: We can conclude from the tables above that none of the other websites above have fully used the three primary contributing human factors (Physical limits of visual acuity, Limits of absolute memory and Gestalt Principle) to design their GUI, resulting in many gaps in the user experience. For example, none of

the sites except for AMPed is designed to handle medium and small screen sizes of tablet and smartphone devices. AMPed responsive design ensures that the data is easily readable on any screen size and maintains data integrity. Amongst all the sites compared, AMPed is the only website that provides a secure login to allow its authorized users to directly contribute to the data. All of the other websites are read only. AMPed's login feature and ability for authorized users to update the data through the website will allow for greater collaboration and easier maintenance of the data by its users.

1.2 Methodology

Agile development methodology is a software development technique in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams [11]. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement, and encourages rapid and flexible response to change.

This project used the agile development principles. The AMPed GUI was created iteratively, doing small but frequent releases. Change is accepted in agile development. In fact, it is expected. Instead of a fixed scope, the timescale is fixed and requirements emerge and evolve as the product is developed. For large and complex databases like AMPed where the user needs vary widely, and often evolve, as new data is available, this approach of developing iteratively provided a way to get the maximum value out of the project in a given timeframe.

The first part of this project focused on identifying high-level requirements and analyzing problems that users encounter when working with the existing database. The requirements were broken-down into the following hierarchy:

- **Capability** *(is a requirement that a product must possess to ultimately satisfy a user need or objective)*
    - **Feature** *(is a set of logically related requirements that allow the user to satisfy an objective or capability)*
        - **Story** *(are short simple descriptions of a feature told from the perspective of the person who desires the new capability. Typically follow a simple template: As a <type of user>, I want <some goal> so that <some reason>)*

This information was compiled into a prioritized requirement document called backlog. The design and style of webpages, functionality, requirements and scenario of the application were described and matched to the user needs. The highest value requirements of the graphical user interface were implemented iteratively. The following diagram depicts project management for agile development that was used in this thesis [11].

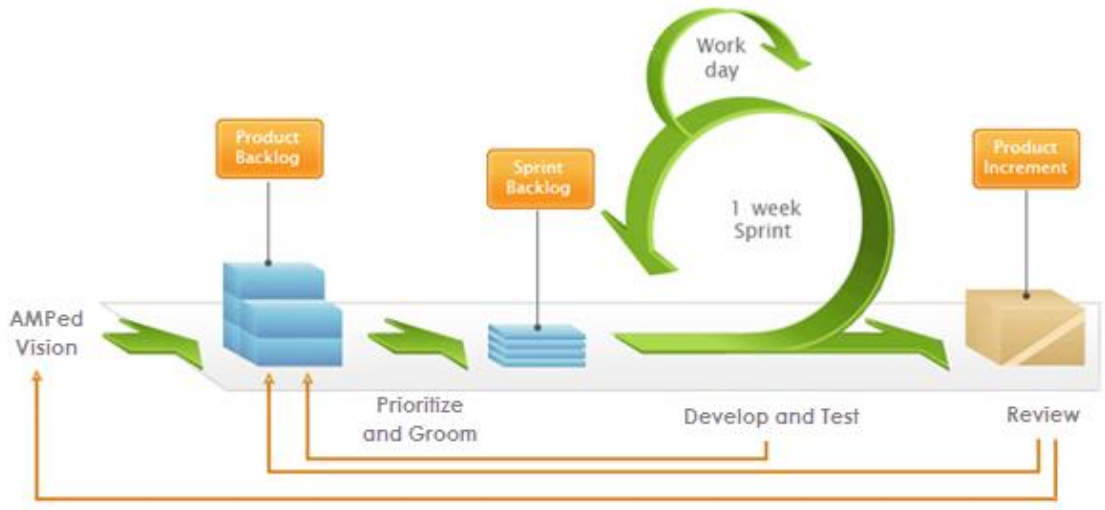


Figure 7: Overview of Agile Process

Now let us take a look at the agile process, define the various phrases in the diagram above and provide a sample of how we used this methodology to develop the AMPed system.

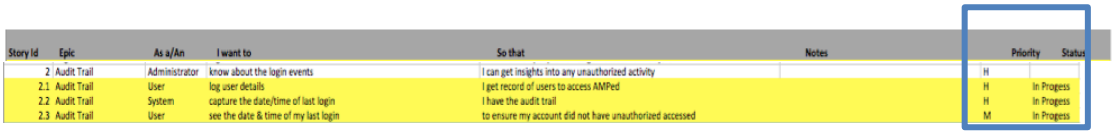
1.2.1. Product Backlog

The Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts customary in software design. These items can have a technical nature or can be user-centric e.g. in the form of user stories. The thesis used Microsoft Excel spreadsheet to capture a list of all the capabilities, features and stories. The sample below shows the login capability (epic) and the associated stories for this epic.

Story Id	Epic	As a/An	I want to	So that
1	Login	User	securely log in	I get access to AMPed update and maintenance features
1.1	Login	Administrator	securely store user id & password in database	authorized users can login
1.3	Login	User	enter my userid	I can loginto the system
1.4	Login	User	enter my password	I can log into the system
1.5	Login	System	validate login credentials	only authorized users access the database
1.6	Login	System	maintain the login session	I do not have to login again and again
1.7	Login	System	timeout the session after certain time	unauthorized people do not gain access to the system

1.2.2. Backlog Prioritization

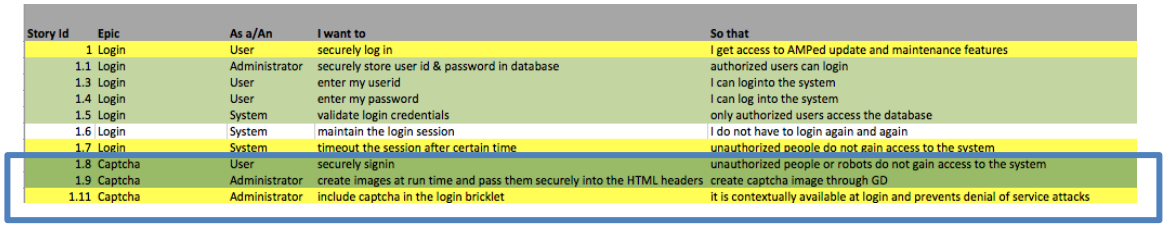
The backlog was then prioritized based on business value and technology complexity. Each story was rated H (high), M (medium) and L (low). This helped to decide the order of pursuit for the various requirements and ensure the highest value capabilities are delivered first. The screenshot below shows how the "Audit Trail" epic and stories were prioritized.



Story Id	Epic	As a/An	I want to	So that	Notes	Priority	Status
2	Audit Trail	Administrator	know about the login events	I can get insights into any unauthorized activity		H	
2.1	Audit Trail	User	log user details	I get record of users to access AMPed		H	In Progess
2.2	Audit Trail	System	capture the date/time of last login	I have the audit trail		H	In Progess
2.3	Audit Trail	User	see the date & time of my last login	to ensure my account did not have unauthorized accessed		M	In Progess

1.2.3. Backlog Grooming

Backlog grooming is a product backlog refinement process that is used to keep the backlog clean and orderly. The AMPed backlog was an evolving document. As new information was discovered, epics/stories were added, dropped and reprioritized. Also, more details were added to epics/stories as the work progressed. The following example shows the addition of Captcha feature to the secure login epic as new information was discovered after we realized that AMPed needed to be secure from denial of service (DOS) attacks.



Story Id	Epic	As a/An	I want to	So that
1	Login	User	securely log in	I get access to AMPed update and maintenance features
1.1	Login	Administrator	securely store user id & password in database	authorized users can login
1.3	Login	User	enter my userid	I can loginto the system
1.4	Login	User	enter my password	I can log into the system
1.5	Login	System	validate login credentials	only authorized users access the database
1.6	Login	System	maintain the login session	I do not have to login again and again
1.7	Login	System	timeout the session after certain time	unauthorized people do not gain access to the system
1.8	Captcha	User	securely signin	unauthorized people or robots do not gain access to the system
1.9	Captcha	Administrator	create images at run time and pass them securely into the HTML headers	create captcha image through GD
1.11	Captcha	Administrator	include captcha in the login bricklet	it is contextually available at login and prevents denial of service attacks

1.2.4. Sprint Backlog

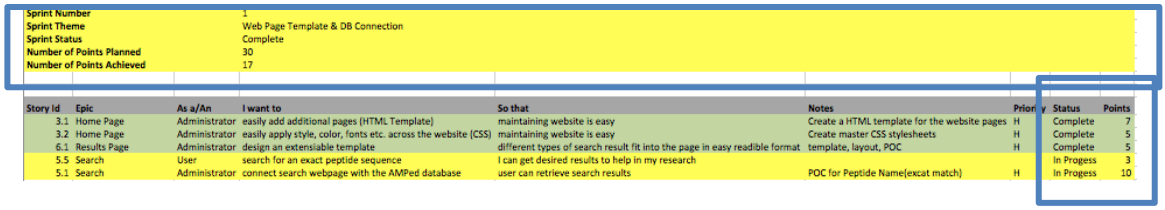
Sprint backlog is simply a list of stories identified by the team (in this case me) to be completed during the sprint or iteration (explained in the next section). At the start of each sprint, I selected a few stories from the product backlog and then identified the tasks necessary to complete each user story. These few selected stories created the sprint backlog. Once the backlog was prioritized, stories were pulled into sprints to create a sprint backlog. Following picture shows a sample backlog of Sprint 1.

Sprint Number	1
Sprint Theme	Web Page Template & DB Connection
Sprint Status	Complete
Number of Points Planned	30
Number of Points Achieved	

Story Id	Epic	As a/An	I want to	So that	Notes	Priority
3.1	Home Page	Administrator	easily add additional pages (HTML Template)	maintaining website is easy	Create a HTML template for the website pages	H
3.2	Home Page	Administrator	easily apply style, color, fonts etc. across the website (CSS)	maintaining website is easy	Create master CSS stylesheets	H
6.1	Results Page	Administrator	design an extensible template	different types of search result fit into the page in easy readible format	template, layout, POC	H
5.5	Search	User	search for an exact peptide sequence	I can get desired results to help in my research		
5.1	Search	Administrator	connect search webpage with the AMPed database	user can retrieve search results	POC for Peptide Name(exact match)	H

1.2.5. Sprints or iterations

Sprint, also known as iteration, is a set period of time during which specific work has to be completed and made ready for review. Sprints contain confined set of work (i.e. sprint backlog as described above) and have a regular, repeatable work cycle. Each sprint for AMPed was one week long. The following picture shows a sample execution of AMPed Sprint 1. The main focus of the sprint was to design an extensible web template for AMPed and connect to the database. The sprint in total had 30 points planned in the backlog. One point was assumed to be roughly equal to three hours of work.



Sprint Number	1
Sprint Theme	Web Page Template & DB Connection
Sprint Status	Complete
Number of Points Planned	30
Number of Points Achieved	17

Story Id	Epic	As a/An	I want to	So that	Notes	Priority	Status	Points
3.1	Home Page	Administrator	easily add additional pages (HTML Template)	maintaining website is easy	Create a HTML template for the website pages	H	Complete	7
3.2	Home Page	Administrator	easily apply style, color, fonts etc. across the website (CSS)	maintaining website is easy	Create master CSS stylesheets	H	Complete	5
6.1	Results Page	Administrator	design an extensiable template	different types of search result fit into the page in easy readible format	template, layout, POC	H	Complete	5
5.5	Search	User	search for an exact peptide sequence	I can get desired results to help in my research			In Progess	3
5.1	Search	Administrator	connect search webpage with the AMPed database	user can retrieve search results	POC for Peptide Name(excat match)	H	In Progess	10

1.2.6. Develop and Test (Sprint Execution)

Sprint execution is like a mini project unto itself wherein all of the work necessary to deliver the stories in the sprint backlog is performed. Stories from the sprint backlog were pulled in one at a time for development in the iteration. Once the development was complete, the testing was done on that story and any bugs were fixed. A story was marked complete once the acceptance criteria were met, i.e. there were no bugs and the functionality matched the desired outcome of the story. Then the next story was picked up for development. As can be seen in the above figure, at the end of Sprint 1, 17 points were achieved, 3 stories were completed and 2 stories were carried forward in Sprint 2.

1.2.7. Review (Sprint Closure)

Sprint review is the last step in this small interment work approach wherein the stakeholders, if available, can see the progress made and review the working code.

Now, let’s take an example of one of the capabilities developed as part of this thesis, Secure Log-in, and describe in detail how agile methodology was used to do adaptive planning, incremental development and early delivery all in tandem with a rapid response to change.

As part of the AMPed vision, Secure Log-in was identified as a very important capability. The Secure Log-in capability was first broken down into the following user stories.

Story Id	Epic	As a/An	I want to	So that
1	Login	User	securely log in	I get access to AMPed update and maintenance features
1.1	Login	Administrator	securely store user id & password in database	authorized users can login
1.3	Login	User	enter my userid	I can loginto the system
1.4	Login	User	enter my password	I can log into the system
1.5	Login	System	validate login credentials	only authorized users access the database
1.6	Login	System	maintain the login session	I do not have to login again and again
1.7	Login	System	timeout the session after certain time	unauthorized people do not gain access to the system
1.8	Captcha	User	securely signin	unauthorized people or robots do not gain access to the system
1.9	Captcha	Administrator	create images at run time and pass them securely into the HTML headers	create captcha image through GD
1.11	Captcha	Administrator	include captcha in the login bricklet	it is contextually available at login and prevents denial of service attacks

All these stories were included in the Product Backlog and prioritized. Then the following three stories were picked for execution and were included in the Sprint Backlog.

1.1	Login	Administrator	securely store user id & password in database	authorized users can login
1.3	Login	User	enter my userid	I can loginto the system
1.4	Login	User	enter my password	I can log into the system

Each sprint was one-week long and the following section details out the work done for the three stories highlighted above.

First, the user story 1.1 was picked and the following tables in the AMPed database were developed:

- User
- Country
- Access_Level

Then, the tables were loaded with some sample data. After the tables were created, the user interface stories (1.3 and 1.4) for entering user id and password were picked. We created the UI templates first, then the data entry fields and then the login submit button. The UI was then tested on various browsers. After this, PHP coding was done to connect the UI with the database tables and tests were done to ensure log-in was successful only for the valid users loaded into the table.

Then the completed work (i.e. the three Login stories referenced above) was reviewed in the weekly meeting with major professors. When the story/stories were approved in the review cycle it was added to the Project increment as final product. In this case, all three stories were approved and added into the login capability for AMPEd.

Any change, enhancement or addition requested during review was taken back to the AMPed vision and Product Backlog. Then the feedback was analyzed to create

user stories that were then groomed and prioritized. Based on the priority, these feedback stories were pulled into the appropriate future sprint backlog for execution.

The three stories above created the following product increment:

This iterative process was used for all the stories for several weeks that in turn guided the completion of AMPed vision. For example, in the next sprint, the following CAPTCHA stories were pulled in for execution.

1.8	Captcha	User	securely signin	unauthorized people or robots do not gain access to the system
1.9	Captcha	Administrator	create images at run time and pass them securely into the HTML headers	Create captcha image through GD
1.11	Captcha	Administrator	include captcha in the login bricklet	it is contextually available at login and prevents denial of service attacks

REFERENCES

[1] Wikipedia is a free encyclopedia, URL: https://en.wikipedia.org/wiki/Database_normalization, Last accessed: 09/20/2015

[2] Codd, E.F., "*Further Normalization of the Data Base Relational Model*". (Presented at Courant Computer Science Symposia Series 6, "Data Base Systems", New York City, May 24–25, 1971.) IBM Research Report RJ909 (August 31, 1971). Republished in Randall J. Rustin (ed.), "Data Base Systems: Courant Computer Science Symposia" Series 6. Prentice-Hall, 1972.

[3] Helander, Martin, "*Handbook of Human-Computer Interaction*", 1988

[4] Jansen, B. J., "*The Graphical User Interface: An Introduction*", SIGCHI Bulletin 30(2), 22-26, 1998

[5] Miller, George A, "*The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. Psychological Review*", Vol.101. No.2:343- 352

[6] Sarna, David E. and George J. Febish, "*What Makes a GUI Work?*" Vol. 4., (July 15 1994)

[7] Wickens, Christopher D., "*Engineering Psychology and Human Performance*", 2nd edition, 1992

[8] APD3, The Antimicrobial Peptide Database, URL: http://aps.unmc.edu/AP/main.php, Last accessed: 2015/03/05

[9] YADAMP, Yet another db of antimicrobial peptides, URL: http://yadamp.unisa.it, Last accessed: 2015/03/05

[10] LAMP: A database linking antimicrobial peptide, URL: http://biotechlab.fudan.edu.cn/database/lamp/, Last accessed: 2015/03/05

[11] Beck, Kent "*Embracing Change with Extreme Programming*". Computer 32 (10): 70–77., 1999, doi:10.1109/2.796139

CHAPTER 2

DATABASE DESIGN AND DEVELOPMENT

The peptide datasets are stored in various open online repositories and databases. These datasets are very complex and are in huge quantity with hundreds of data entries for each peptide protein. Biology researchers often need to search through these numerous sources of very large quantities of duplicative data. This makes peptide data access a very complex and time-consuming task. Many individual researchers, experimenting with different strategies, have built their own local databases adding to an ever-increasing volume of peptide data. Presently, there are several large publicly available databases like UniProt and NCBI [1][2]. But none of them uniformly annotate their result that makes correlating entries for identical proteins or peptides from one database to another generally a Herculean task.

The idea behind "AMPed" (Anti-Microbial Peptide Editable Database) is to create an annotated collection of antimicrobial peptides that are sourced from several online repositories, journals and existing large databases such as ATCC, PDB, UniProt and NCBI with the purpose of uniformity and coherence [2][3][4].

The database is developed using MySQL, an open source database and Relational Database Management System (RDBMS). AMPed now has a simple and secure web interface that researchers can use to find and download sequences relevant to their research while easily maintaining links to the original data source [5]. For this thesis, we will call earlier work on AMPed "Version 1" and this thesis work "Version 2". The

analysis and research done in this thesis discovered that some areas of the Version 1 database needed to be updated and reorganized. There was some duplicity and the database was not fully normalized. Database normalization is the process of organizing the columns (attributes) and tables (relations) of a relational database to minimize data redundancy and prevent anomalies in the data results. The objective is to isolate data so that additions, deletions, and modifications of an attribute can be made in just one table and then propagated without error through the rest of the database using the defined foreign keys used to connect the database tables [5][6].

The AMPed web interface depends on the data in the AMPed database to provide users an insight into the different aspects of antimicrobial peptides. To improve this capability and to reduce potential inconsistencies in the data, the AMPed database was redesigned and implemented again.

To redesign the AMPed database, I first updated the entity-relationship (ER) model of Version 1. This is conceptual model of the database. We looked at the various user needs and added/deleted entities and relationships to refine the model further. This model was then mapped to a new database schema, AMPed Version 2. Since the peptide data is very complex and available in bulk, new information to improve the design was discovered on an ongoing basis. Also, different researchers had different needs that were discovered as the thesis progressed. So the database design kept changing as more information and new aspects were discovered and added. We used agile iterative development process to respond to the changes with constant communications and inputs from the thesis major professors; the final design

of the database was then completed. Throughout the process the ER diagram was first modified, which in turn guided the database schema or table specification process. [7]

The Version 1 of the AMPed database diagram is given in Figure 8 for reference.

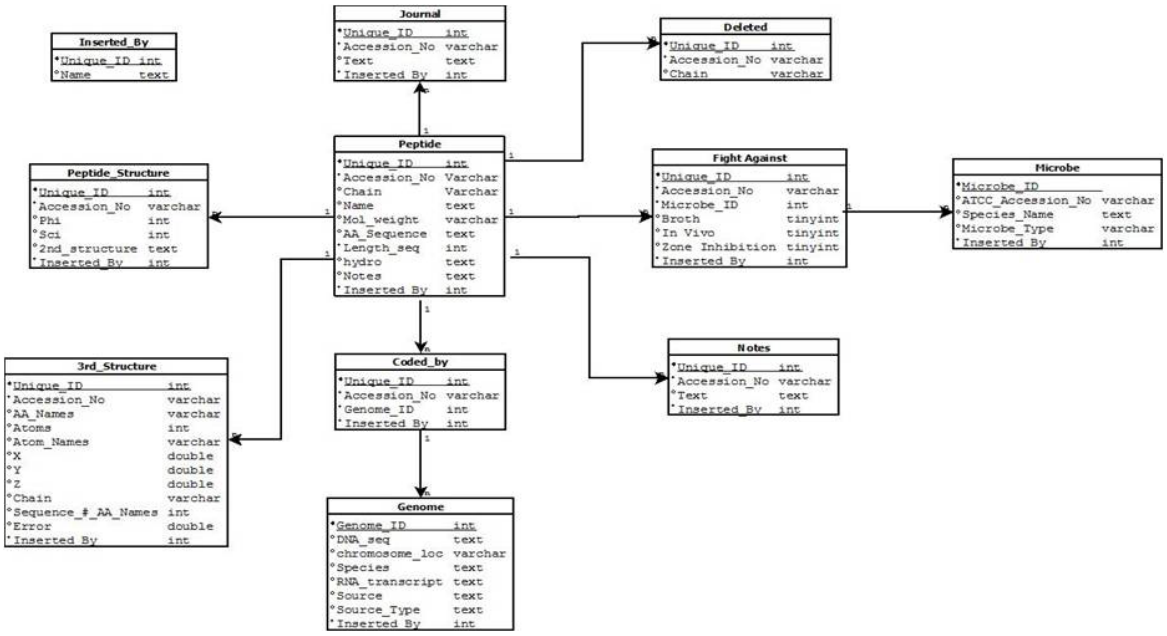


Figure 8: Version 1 of the AMPed Database

Many major and minor changes made along the way are described further:

1. **Eliminate Data Redundancy:** In Version 1, the attributes Accession_No and Inserted_By appeared in almost all entities leading to data redundancy. This sort of redundancy could harm the integrity of data if one were to update, for example, Accession_No or Inserted_By. To update these fields in Version 1, queries would have to be run on multiple tables, which would have been time consuming and error prone.

2. **Remove Invalid Data Entities:** Deleted and Notes entities are related to the logic of managing the Bioparser and not directly needed to manage the peptide data. After consultation, these were removed from the AMPed database.
3. **Rationalize Data Formats:** The format of the attribute Microbe_ID, was changed to give global recognition and information from the AMPed database. The data types of attribute Microbe_ID were changed from “int” to “varchar”, as is appropriate to the data.
4. **Renamed Attributes:** Attribute names in entities like Genome were changed to ensure they are more globally recognized and align with the biological names that researchers use. The attribute Unique_ID is changed to AMP_ID to avoid the confusion with other ID attributes in the database and to indicate the table in which this is a unique identifier. The attribute Sequence_#_AA_Names is changed to Sequence_AA_No because MySQL queries are not compatible with attributes having special characters.
5. **Add Missing Data & Entities: A** few new entities were added to correctly support the desired biological information and to secure the AMPed database. For example, Journal entity was changed to Article and was enhanced to contain more information about any publications about the peptide, not just journal articles. The Peptide_Structure entity was changed to Atomic_Coordiantes and new attributes were added. 3rd_Structure was changed to 3D_Structure with new attributes. A new entity User was added to maintain user data and access levels.

6. **Normalize Entity Relationships:** The entity relationships in the database were redone to align with the newly created entities and the data within them. For example, the attributes of entities Peptide, Fight_Against, 3rd_Structure were broken down into new tables with one-to-many relationships.
7. **Cardinality of Relationship:** Some relationships earlier identified as 1:1(one to one) or 1:N (one to many) were changed to higher order cardinalities to be consistent with the nature of the data. For example, the number of occurrences in Peptide entity associated to the number of occurrences in Article entity is changed to N: M (many to many) relationship.

The entity relationship diagram in Figure 9 shows the new relationships of entity sets stored in the AMPed database.

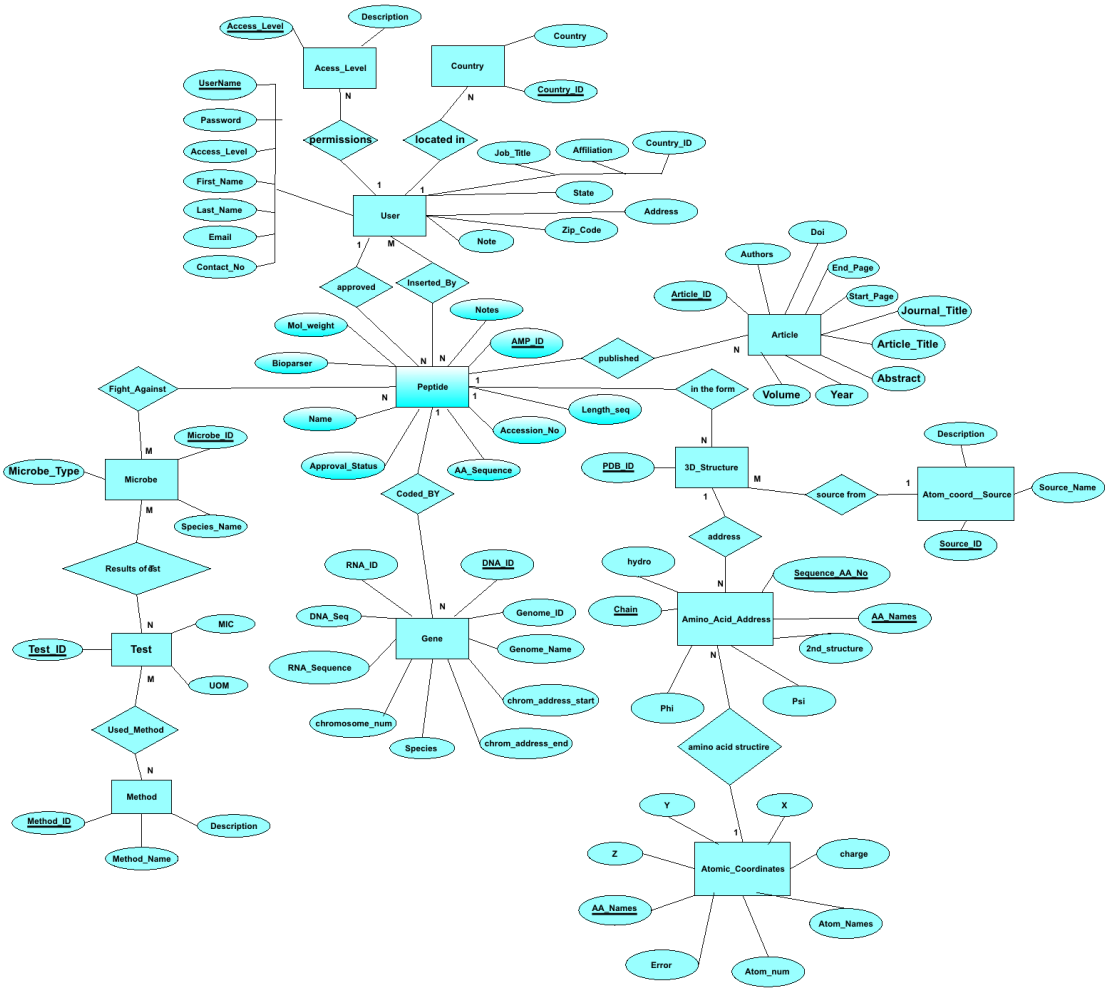


Figure 9: Entity-Relationship Diagram of Version 2 AMPed Database

The ER diagram (Figure 9) pulled together all the entities and relationships of the AMPed database in one unified diagram. Each entity appears on the diagram just once and connected to one or several other entities by lines that explain the relationship between each pair. It helps to better organize the primary data concepts and entities of the AMPed. This ER diagram also shows how everything comes together. For example, looking at the ER diagram:

1. It can be easily determined that the main core table is Peptide, which shares relationship with almost every table.
2. The Peptide fights against the Microbes have one-to-many relationship between them.
3. Microbes are tested by many different methods like In Vivo, Broth. They share many-to-many relationship.
4. The peptides can be coded by the Genome.
5. Peptide is in the 3D structure form. The values of 3D_Structure can be calculated by many different sources like X-ray, NMR.
6. The 3D_Structure's measurements of the location of amino are stored in Atomic_Coordiantes. This is one-to-many relationship.
7. The information about the peptide can be collected/referred from published articles. The records of Peptide can be Inserted_By User. The User granted permission by Access_Level and User can be located in any Country.
8. The new entity User, Method, Test, Atom_coord_Source, Access_Level and Country are added which defines new relationships among the entity sets. This helps meet new requirements where in users are required to be members to access AMPed.

Each entity of the AMPed ER model is described in Table 1:

Entity	Description
Peptide	The basic information about peptides like accession number, name of peptide, amino acid sequence is stored
Microbe	Information about the microbe and species
Test	Result of test performed using microbe and peptide like MIC, Unit of Measurement
Method	Method used for the test such as Broth, In Vivo
Gene	Information about the host, DNA sequence, chromosome location
3D_Structure	Describes 3D structure of peptides
Amino_Acid_Address	Information about amino acid location like chain, length sequence
Atomic _Coordinates	Atomic coordinate values such as X, Y, Z coordinates
Atom_coord_Source	The source used to scan the structure of the peptide
Article	Information about the article referred for the peptide
User	Information about the user granted permission for AMPed
Access_Level	Defines what permissions are granted to the user
Country	The user or his/her affiliation country location

Table1: Description of Entities of the AMPed ER model

Entities and their relationships as described in the ER diagram above were built iteratively. The ER diagram helped to discover different relationships among entities

and helped to communicate the one-to-many and many-to-many relationships that exist in the dataset. The ER diagram also illustrates the logical structure of the AMPed database and captures the conceptual structure of the database. This ER diagram has been converted into relational database schema. Relational database schema is the skeleton structure of AMPed database. It defines how the AMPed data is organized and how the relations are associated amongst the data. It formulates all the constraints that are to be applied on the data. Each entity of the AMPed ER diagram was mapped to the database tables and attributes to columns. The key attributes became the primary key of their respected tables.

The new AMPed database schema created in this thesis is normalized to third normal form (3NF). Each table contains only information that pertains to that table. Any information that does not pertain to it is moved into another appropriate table, which assures non-redundant storage, and maintains the desired data integrity constraints. If one of the rows that are part of a reference is changed, all the references to it will be updated immediately. Thus, the 3NF AMPed database minimizes data redundancy, helps to maintain data integrity and supports the construction of robust search queries.

If it is necessary, to add or change the AMPed database design in the future, the ER diagram here can be referenced to assist understanding of the logic structure of the database, saving considerable time and effort.

Database Schema of AMPed

The diagram (Figure10) below shows the current AMPed database with tables, cross-referenced tables, primary keys and constraints applied. It also describes the physical structure of the AMPed database.

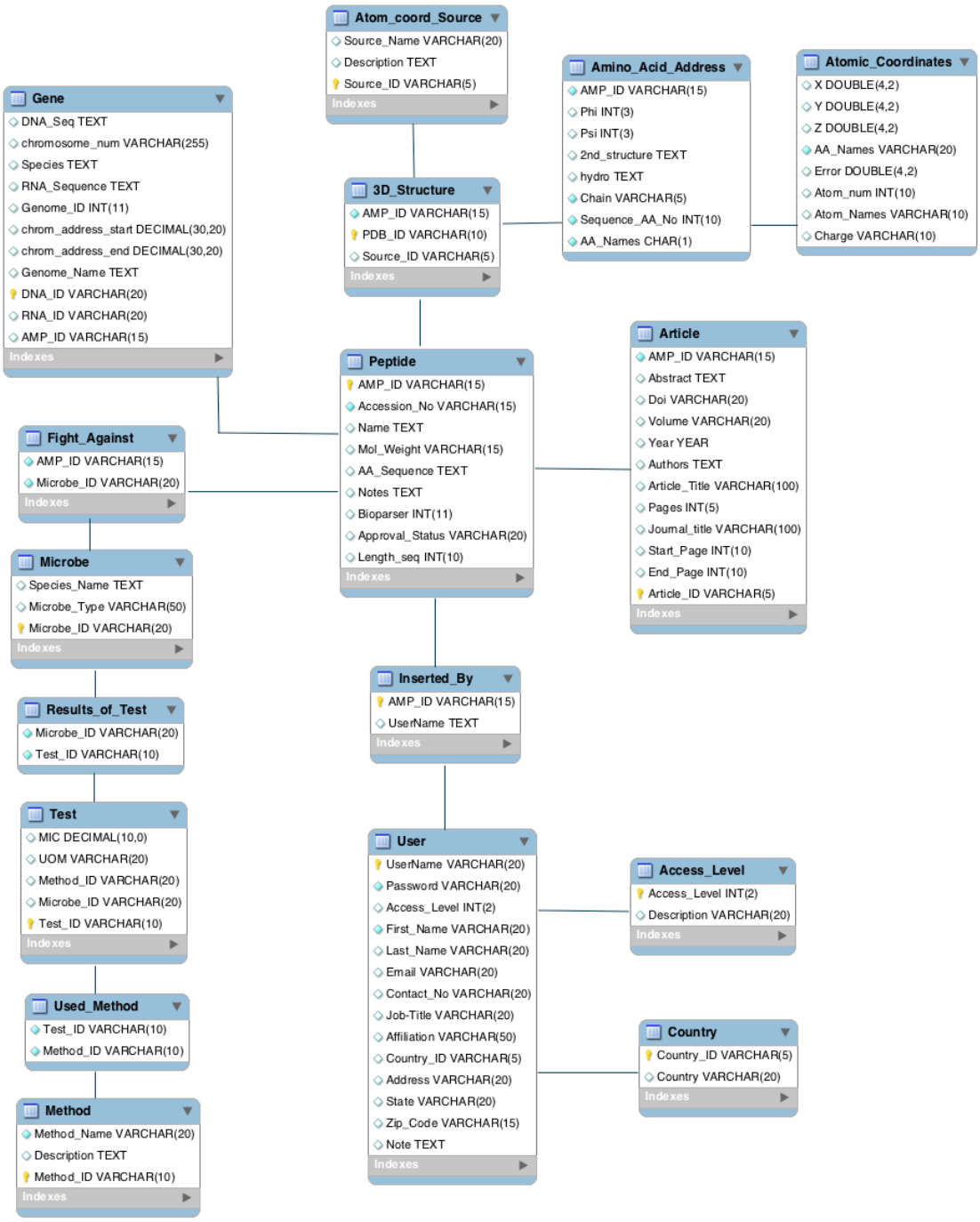


Figure 10: Tables of Current AMPed Database

Legends for Figure10

 No Constraint

 Not Null

 Primary Key

The description of the AMPed database tables with its attributes, data types and details are given in appendix. As we completed the database schema, we were able to design and implement the interface. The interface helps the user to access the data and cannot be implemented until the structure of the data is settled.

REFERENCE

[1] UniProt (Universal Protein Database) is a central repository of information on proteins, URL: http://www.uniprot.org, Last accessed: 10/28/2015

[2] NCBI (National Center for Biotechnology Information) provides access to biomedical and genomic information, URL: http://www.ncbi.nlm.nih.gov, Last accessed: 10/28/2015

[3] ATCC (American Type Culture Collection) worldwide repository and distribution center for cultures of microorganisms, URL: http://www.atcc.org, Last accessed: 10/28/2015

[4] PDB (Protein Data Bank) is a crystallographic database for the three-dimensional structural data of large biological molecules, such as proteins and nucleic acids, URL: http://www.rcsb.org, Last accessed: 10/2/2015

[5] Konstantinidis George, Thesis on the design and populating of the AMPed database "*Antimicrobial Peptide Editable Database*", 2015

[6] Wikipedia is a free encyclopedia, URL: https://en.wikipedia.org/wiki/Database_normalization, Last accessed: 09/20/2015

[7] Codd, E.F., "*Further Normalization of the Data Base Relational Model*". (Presented at Courant Computer Science Symposia Series 6, "Data Base Systems", New York City, May 24–25, 1971.) IBM Research Report RJ909 (August 31,

1971). Republished in Randall J. Rustin (ed.), “Data Base Systems: Courant Computer Science Symposia” Series 6. Prentice-Hall, 1972.

CHAPTER 3

WEB INTERACTION DIAGRAM

There are many factors that determine the process of designing and building a website, in particular, the scope of the project, the general size of the site, the level of complexity and other functional requirements. These factors are widespread and diverse leading to several challenges in documenting how the web pages interact with each other and what they contain. As part of this thesis, we looked at several techniques like web flow diagrams, site map diagrams and web template designs. However, none of them individually were sufficient to paint a full picture of the web interactions. Large-scale web development projects, like AMPed, present a unique documentation challenge, especially related to visually documenting these varied web interactions. The work here specifically provides solutions for the following:

- Specific web design strategy for this and similar applications
- Web programming approach for this and similar projects.

A web flow diagram is basically a flowchart that paints a picture of all the pages in a website and how at a high level they are connected and grouped with each other. It lacks the details around content on the page or any interactive features the pages may have. Site maps are similar and provide more detailed view of the cross-connections and groups but no functionality and content. Web templates are great in describing content on the page and the various elements but they lack the details on re-

usable components and the cross-connections and groups. Since diagrams are an essential tool for communicating information architecture and interaction design when developing website, this thesis created a new technique to document the website interaction. We called it “Web Interaction Diagram” or “WID”. The following section discusses the considerations in development of WID and outlines the basic symbols we used for diagramming information and interaction design concepts. It also provides guidelines for the use of these elements. The diagram here serves as a touchstone document for the development of more detailed documents specific to the needs for future web development projects.

WID is based on Christopher Butler’s work, “The Strategic Web Designer”, and Jesse James Garrett work, “A visual vocabulary for describing information architecture and interaction design” [1] [2]. WID uses many visual vocabulary elements from James work to describe a system, structure, reusable processes, interactive designs and the flow of the web site. WID and its descriptions can be used to map the following:

- Visual and interface designs
- Unique pages and a high level navigational and interface designs
- Information architecture of the page and content elements
- Interaction designs used to develop each page

This thesis considered the following key things, based on Jesse James Garrett work, when building a WID and its visual vocabulary:

- **Simple and distinct:** The vocabulary should be simple enough that diagrams can be sketched quickly by hand. The elements of the vocabulary

should be distinct enough from each other and there is the clarity of the diagram. [1][2]

- **Tool-independent:** The vocabulary should be designed so that specialized software tools are not required in order to construct diagrams. The vocabulary should enable developer to work with the tools they are most comfortable using. [1][2]
- **Small and self-contained:** The diagrams may be used by a diverse range of users with different levels of knowledge of diagramming systems. So, the vocabulary should not require technical knowledge. The total set of elements should be kept as small as possible, maintaining a strict one-to-one correlation between concepts and symbols, so that the vocabulary can be learned and applied quickly. [1][2]

To develop WIDs, we combined the visual vocabulary principles above with traditional software design techniques. The following section describes the three-phased process that can be used to design a WID for any web development project:

- **Phase 1 – WID Information Gathering:** This initial phase of analyzing the sequence and flow of website may not be necessary for every project. However, many situations require the developer to be focused on goals, and given these goals, work through a strategic and organized approach to the website development. In general, it is important to discuss the purpose and the goals of the website, collect requirements, specify the capacity of the system collecting data, discuss platform. This helps to assemble a big-

picture view of the scope of the project based upon this information. When we design the WID, we start with a vision, and then set up rules that everyone else involved in creating content from that point forward could follow. The initial design creates a high-level structure scope of the WID. High level structure of WID will contain a list of unique website pages and a high level navigational flow.

- **Phase 2 – WID Layout Template:** The WID continues to build upon decisions made in the design prototype process. In this phase, as the design process focuses on how the visual presentation reinforces the purpose of the website while also clearly communicating the information, WID captures the data and visualizes the information architecture. Once the initial list of website pages, navigational flow and data layout is cleared, a WID layout template is set.
- **Phase 3 – WID Build & Complete:** In this phase, we add the inputs from visual prototyping for the web (like elements, color palette, typography, texture in a context, which type styles, sizes and edge treatment of images and other details, such as buttons and spacing) into the WID. Also, as the content and the dynamic/interactive features of the website are developed, they are added into the WID, thereby completing the WID.

In AMPed, the objective of describing the WID is to emphasize how the user flows through defined tasks, and what the discrete steps are within these tasks. As described earlier, WID displays navigation along with the information about interface.

WID vocabulary is based on a simple conceptual model on interaction design. Based on the above concept the following web interaction diagram (WID) depicts the flow of the AMPed website. The diagram is created using word document tools.

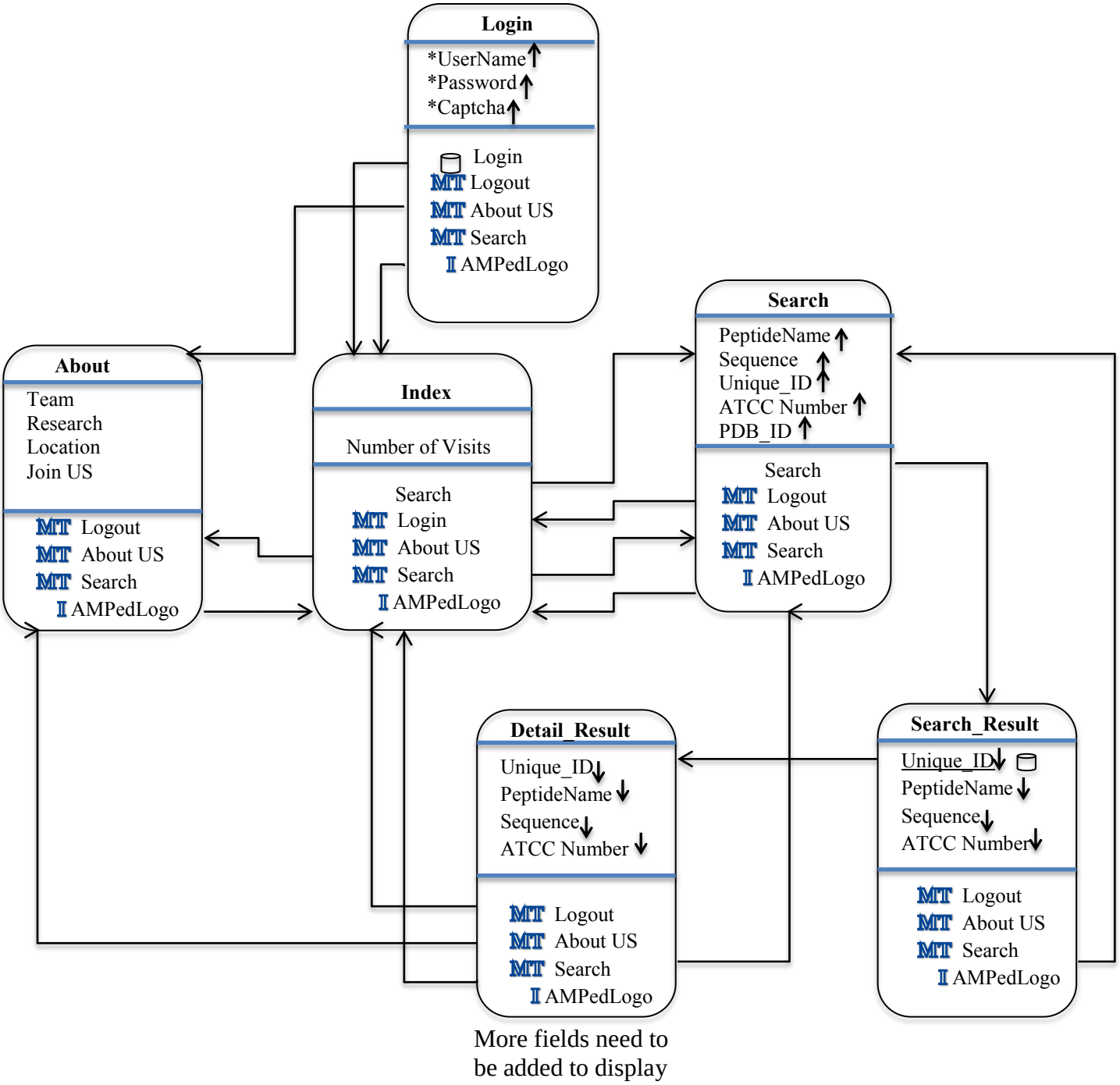


Figure 11: AMPed Web Interaction Diagram

Pages

The basic unit of user experience on the web is the page, which is represented here as a simple rectangle.

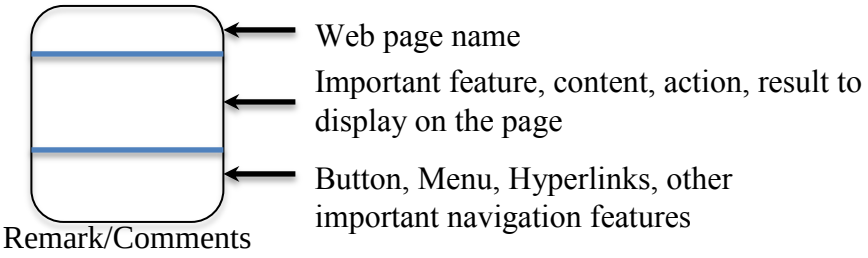


Figure 12: Basic Unit of Web Interaction Diagram

Description of the symbols

* Mandatory

↑ Input by User

↓ Output from the system

Interaction with Database

I Image hyperlinked

M Navigation through Menu

T Text Hyperlinked

Upload

Download

Letter/word/digit Output from the system hyperlinked

Connectors and Arrows

Navigations between pages are depicted with simple arrows or connectors. But all navigational relationships in the diagram may not be noted.

Here, connectors also need to convey directionality to indicate how the user will move through the site toward completion of a particular task. So, arrows will do the trick nicely.

These arrows are not like indicating a one-way street, but rather indicating the way to move forward. The user is not prohibited from moving in the opposite direction; the arrow merely indicates the direction in which the user is likely to want to go.

But there are some actions that require user couldn't go back i.e. irreversible actions examples delete records, place the payment etc. Such navigation can be indicating by using the crossbar on the end of the arrow.

As the user move from one page to another, the previous page can remain open in the browser or closed. In the web interactive diagram, by default, the previous page is assumed to be close. Otherwise, the previous page should remain open and having one on the end of the arrow can indicate this.

AMPed WID figure is described in detail below to explain the level and types of information we can capture:

- **Index Page:** Number of Visits is the main feature; Search is the main button; Search navigates to Search Page; the Logo of the AMPed is in clickable image form and opens the Index page; the user can navigate to Search Page either from Menu or Search button on the Index page.
- **Log-in Page:** When Login from menu is clicked, the page Login opens and Index page is closed; UserName, Password and Captcha fields are mandatory inputs; the login button on the page when clicked interacts with the database.
- **Search Page:** Search Page has contents like PeptideName, Sequence, Unique_ID, PDB_ID, Sequence that can be inputted by the user; the Search button when clicked interacts with database and opens Search_Result;
- **Search Results Page:** Unique_ID, Peptide, Sequence, ATCC Number are outputs given by the system and displayed on results page; the Unique_ID interacts with database and is hyperlinked; when Unique_ID is clicked, the Detail_Result page opens.
- **Detail Search Results Page:** The Detail_Result has Unique_ID, ATCC Number, Peptide fields.
- **Menu Bar:** Login, About US, Search is part of menu in text form and hyperlinked to their respective pages; all pages have same menu and logo

functions, except Login, which is changed to Logout after user logged in the system.

- **About Us Page:** User navigates to About US page through the menu About Us. The content of About US page are Team, Research, Location and Join Us.

Benefits of WID

WID provides the following benefits:

- Identifies a group of pages that share one or more common attributes. E.g. it can be seen very clearly from the figure that the menus of all pages are similar and have common properties.
- Gives clear understanding and idea of the design and the content on the page. A site map simply shows an outline of the pages a website will contain but misses out on the next level of detail that this diagram helps to fill.
- Gives more insights into what technologies and techniques are used on a page and if you updates or modify the project, what may be required e.g. database connectivity, SQL Queries, CSS etc.

In a typical waterfall project, this level of detail is documented in Technical Specification Documents (TSD) but since agile is documentation light and does not typically produce TSDs, this diagram can help serve the longer term needs for

understanding the site. It also will help avoid big reverse engineering code projects when future enhancements are needed.

REFERENCE

[1] Butler Christopher, "*The Strategic Web Designer*", Publisher: HOW Books, August 22, 2012.

[2] Jesse James Garrett, "*A visual vocabulary for describing information architecture and interaction design*", URL: http://jjg.net/ia/visvocab/, Last accessed: 11/2/2015

CHAPTER 4

WEB INTERFACE: Design, Development, Search, Secure Access and Audit Trail

The newer part of the AMPed project, specifically, builds an easy, secure and intuitive web user interface suitable for novice and expert researchers to efficiently search, display, manipulate and eventually upload data into AMPed. The user interface not only provides insights into various aspects of the data set, but it also gives a visual presentation of the available data when needed. The AMPed user interface can be easily accessed through desktop, laptop or smartphone devices via a web browser.

Before we dive deeper into the AMPed user interface, we will take a closer look at its architecture. Looking at the architecture will help us to understand the structure of the APMed web system from different perspectives and how we managed the complexity of AMPed in the most efficient and understandable way.

4.1. Architecture

AMPed web is based on a two-tier client-server architecture. Client-server architecture is a network architecture in which each computer or process on the network is either a client or a server. Servers are powerful computers or processes dedicated to managing disk drives (file servers), printers (print servers), or network traffic (network servers). Clients are PCs, laptops, tablets or smartphones on which users run applications. Clients rely on servers for resources, such as data files and processing power [1]. In AMPed, researchers primarily use their laptops and PCs to request the peptide data. A webserver receives the incoming requests from the

researchers' machines, validates the requests and then does the necessary processing to return the requested services (i.e. data and web pages). The Figure 13 below shows the client-server architecture of the AMPed web.

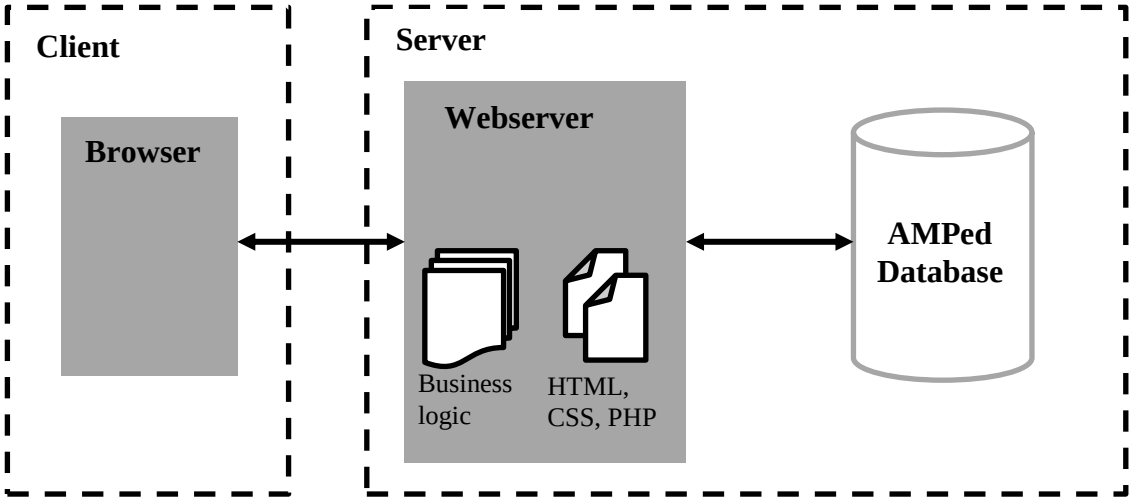


Figure 13: Client-Server Architecture of AMPed

AMPed's client-server architecture, as can be seen in the figure above, has three major components:

1. **Client Side:** Client side is the actual interface for the user of AMPed website. The application such as web browser (Firefox, Safari, Google Chrome, Internet Explorer etc.) on the client machine sends service request data to APACHE webserver running on a powerful server machine at URI. The webserver then either sends an existing page to the client machine or generates a new page and sends to the client machine accordingly. On the client side, the web browser displays the web page, constructed by using the HTML, CSS and Bootstrap sent by the webserver.

2. **Server Side:** Server side is the logical controlling part of the AMPed website. The web container Apache running under the server machine handles the client request, validates with the server side program written in PHP and then generates an appropriate page or locates an existing appropriate page and sends that page to the client side.
3. **Database:** Database is at the back end of the client-server architecture. The data stored in the database is gathered, organized and designed in a sophisticated logical manner using RDBMS and stored in multiple tables. The webserver pulls up data with the help of a database server MySQL, fit it into a web page and then sends it to the client machine.

Now that we understand how the peptide data is stored, processed and viewed by the researchers, let's take a deep dive into AMPed user interface design. Let's start with the webpage layout that explains how AMPed web interface is built and how it looks to the end user.

4.2. Webpage Layout

The AMPed user interface has an intuitive design that makes the information easily accessible to researchers and it can be accessed through desktop, laptop or smartphone devices via a web browser. To develop the AMPed GUI, this thesis combined technology, cognitive science, human need and the latest web design practices. The modular components and screens developed here will provide a good model for other similar web based GUI for databases. In addition, the agile

development principles used by this project helped the research team to organize and prioritize their requirements and helped to iteratively add features to the GUI as user needs evolved.

The AMPed webpage has a simple and extensible layout. Each page has visible branding, persistent navigation and intuitive content location organized for simple but user empowering functionality. Behind the scenes, each page uses a defined template that has externalized styles and reusable components built for scale and performance.

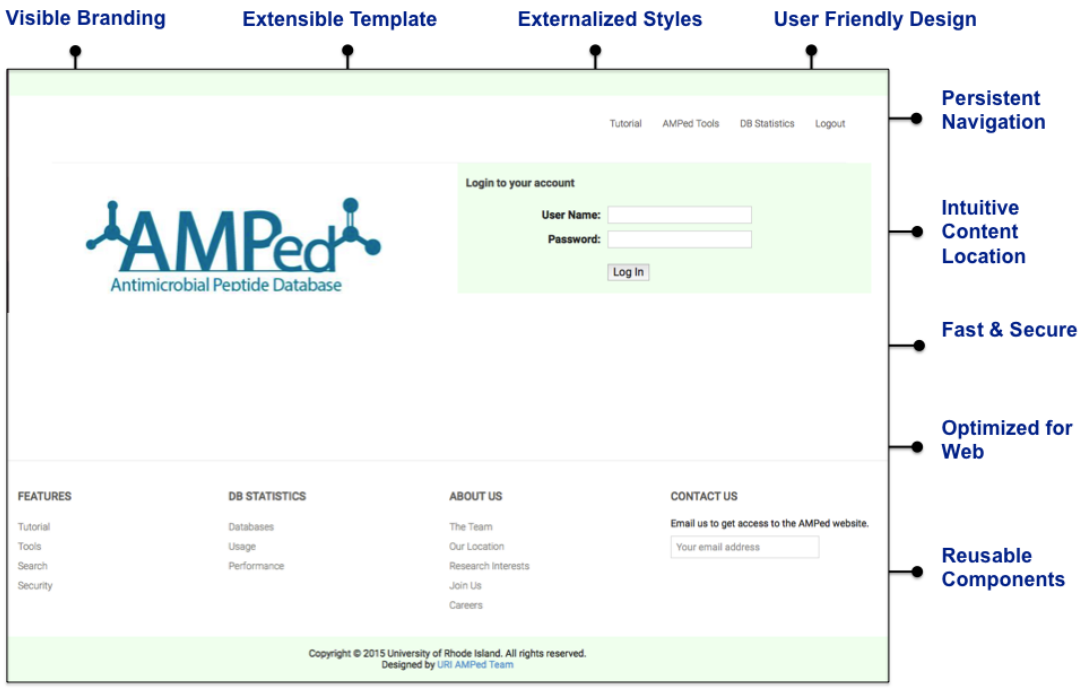


Figure 14: AMPed Web Layout

The section below provides insights into few of these layout features.

4.2.1. Reusable Components

To achieve fast and less error prone development for edits and enhancements, this thesis built AMPed webpages using a component driven design. In order to achieve this, each page, capability and feature was analyzed to check its use and reuse on the site. For example, web Footer was identified as content that will be used at multiple pages and may need to be updated every year for copyright information or every time the navigation is enhanced. Thus the footer was built as a standalone web page component that was included (via PHP's include command) in each page of AMPed website. The figure below shows the webpage footer and the associated include command.

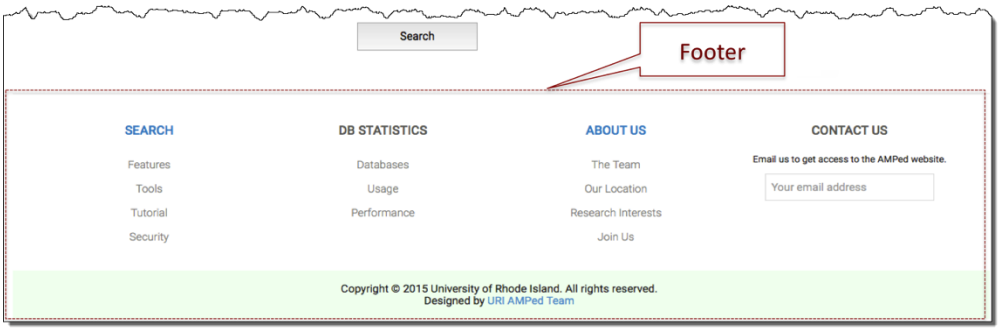


Figure 15: AMPed Footer

Sample of the code used in all web pages to include the file footer content.

```
<?php
        include("footer.php");
?>
```

4.2.2. Extensible Templates

Each page in AMPed is built on templates using DIV tags. The DIV is a generic block-level element. It doesn't convey any meaning about its contents. It is easy to

customize according to varied design needs. The DIV element is currently the most common method for identifying the structural sections of a document and for laying out a web page using CSS. It is used as an “anything-goes” element. It can contain inline or block-level elements. So, it can contain almost any other element. Also, this element has no compatibility issues. Almost all the devices, browsers and those listed in this thesis document support the DIV element.

The DIV element in AMPed is used to group varied content and functional elements together. The HTML code snippet below shows DIV tags being used to identify different sections of the AMPed web page and in conjunction with html attributes. Detailed code is also given in the appendix for reference.

```
<div class="col-sm-8">
        <div class="shop-menu pull-right">
           <ul class="nav navbar-nav">
             <li><a href="search.html">Search</a></li>
             <li><a href="#">AMPed Tools</a></li>
             <li><a href="aboutus.html">About Us</a></li>
             <li><a href="#">Login</a></li>
           </ul>
        </div>
</div>
```

4.2.3. Externalized Styles

This thesis used external style sheets to define styles for AMPed web pages, including the design, layout and variations in display for different devices and screen sizes. These external style sheets were stored in Cascading Style Sheet (CSS) files. CSS is a stylesheet language that describes the presentation of a web page in HTML. CSS describes how elements must be rendered on screen, on device, or in other media. CSS saves a lot of work as it can control the layout of multiple Web pages all at once.

A CSS rule set consists of a selector and a declaration block as in the figure below:

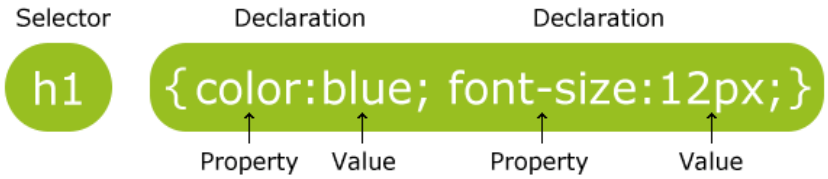


Figure 16: CSS Rule Example

The selector points to the HTML element that one wants to style. The declaration block contains one or more declarations separated by semicolons. Each declaration includes a property name and a value, separated by a colon.

Here is a small snippet of the AMPed code that illustrates the use of CSS. In the following example, main.css file was included in the AMPed webpage. The code illustrates how a part of this CSS styles with all <h> elements will have font-family Roboto with sans-serif.

```
h1, h2, h3, h4, h5, h6 { font-family: 'Roboto', sans-serif; }
```

4.3. AMPed Web Flow

The following figure of AMPed web flow chart gives a big picture of the pages that the AMPed website contains in a high level site map.

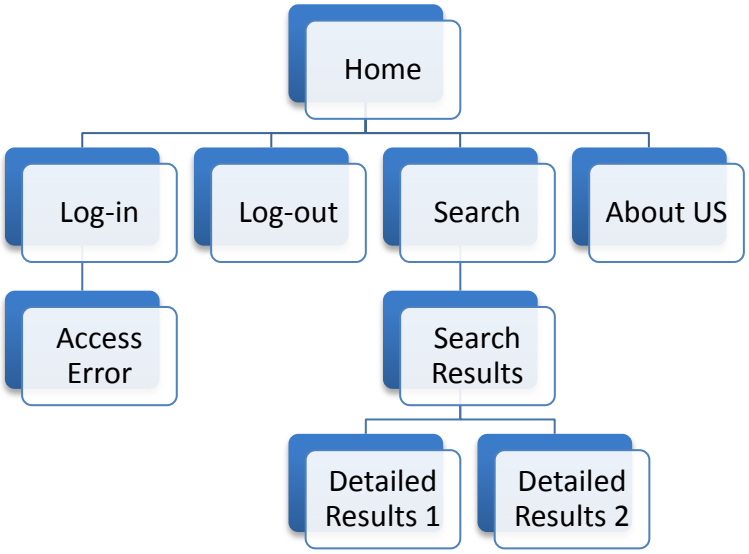


Figure 17: AMPed Web Flow Diagram

4.3.1. Home (Index) page

The home page provides an overview of the AMPed tool and a summary of all the varied features it has. Among other things, home page provides a quick access to search feature and also displays the number of visits (web hit counter) on the AMPed website.

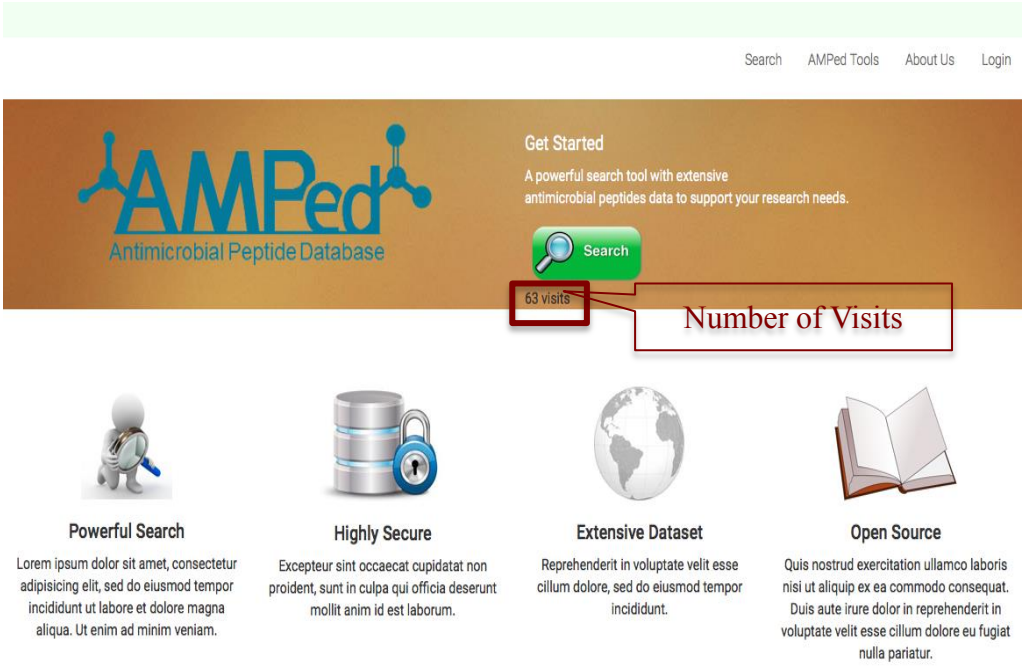


Figure 18: AMPed Home Page

4.3.2. Web Visit Counter

The number of visits as shown in Figure 18 track website the number of visits to the AMPed website. A visit is one individual visitor who arrives at AMPed web site and proceeds to browse. A visit counts all visitors, no matter how many times the same visitor may have been to the site. The file hit_number.php code tracks the number of each visit to AMPed web page and writes the number of visits in a text file Countlog.txt.

4.3.3. Login Page

The login page is used for entering identifier information by a user in order to access the AMPed data. The login function in AMPed requires the user to enter three pieces of information:

1. **User Name:** referred to as an account name, is a string (i.e., sequence of characters) that uniquely identifies a user. User name in AMPed is a completely arbitrary value assigned by the AMPed administrator.
2. **Password:** similar to User Name, password is a string, but it differs from a user name in that it is intended to be known only to its user. Passwords do not display in clear text in AMPed GUI.
3. **CAPTCHA:** is a graphic presented with distorted text used to tell whether the user is a human or a computer. It is described in detail later.

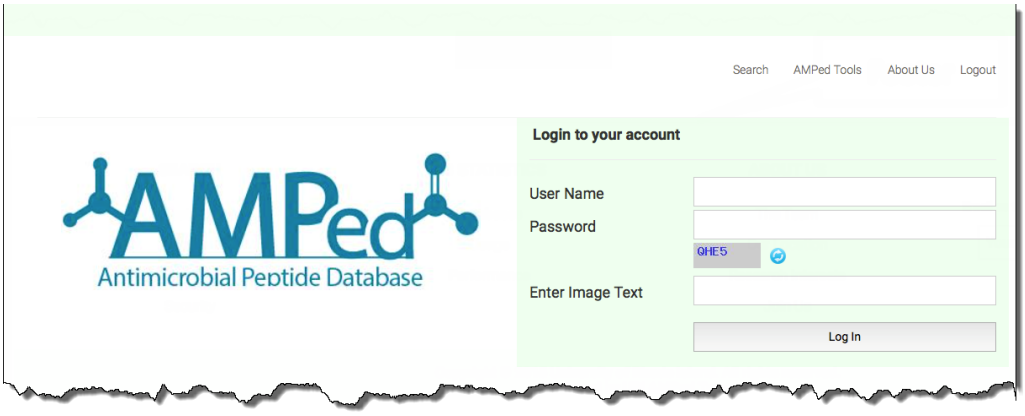


Figure 19: AMPed Log-in Page

These three pieces of information are entered into a login bricklet on the AMPed GUI shown above. When the user attempts to log into the system, a Captcha image is first generated by the system and compared to the text entered by the user. If correct, the user names and passwords entered by the user are compared with data contained in special User tables in the AMPed database. If successful, user is provided access to the AMPed site. The process of logging in creates a session (i.e., a period of use), also referred to as a login session, for the user on the AMPed system. The user is able to

access AMPed secure content only while the login session is alive. In addition to restricting access, logins also provide an audit trail in the form of data that is automatically entered into system log files (i.e., automatically updated files that contain records of events that have occurred on a system).

4.3.4. About Us

The About Us page provides an overview of the team behind the AMPed effort. It also provides insights into the research work that the team is doing.

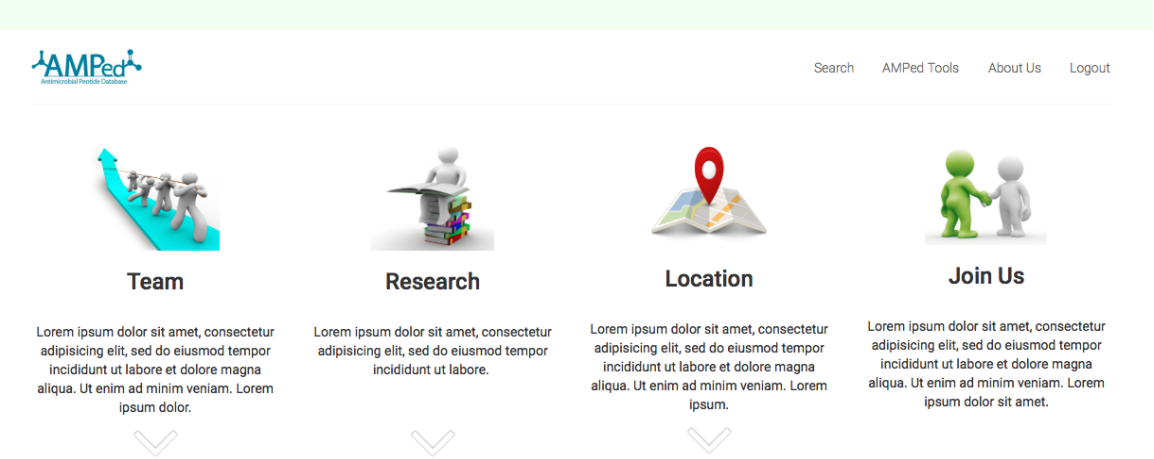


Figure 20: AMPed About Us Page

About Us page also provides an interactive location map built using Google maps called inside of a frameless iframe. The iframe HTML element is often used to insert content from another source into a web page. The HTML document of Google maps is

embedded inside aboutus.html document using <iFrame> tag. There is no border bound to this iframe tag.

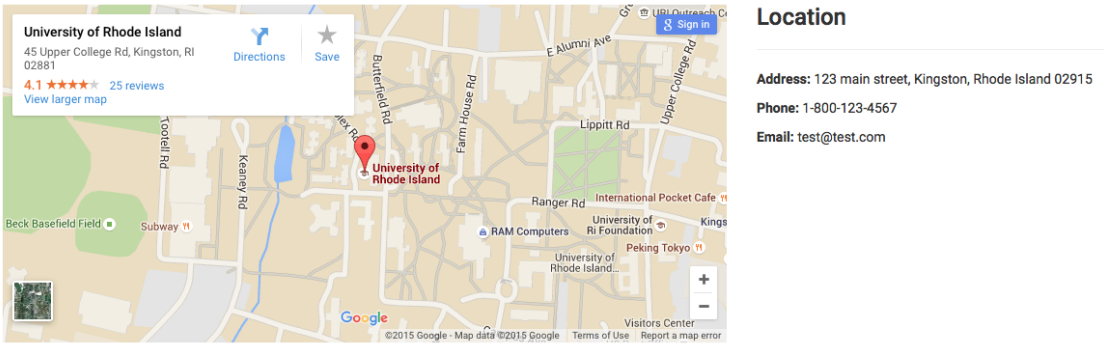


Figure 21: AMPed Location Map Page

4.3.5. Search Criteria

The search criteria page allows a user to search for specific information in the AMPed database. As can be seen in Figure 22 below, a user can pick and choose one or more items to search for. Based on user selection, AMPed, behind the scenes, formulates the search query and processes the data to display summary results.

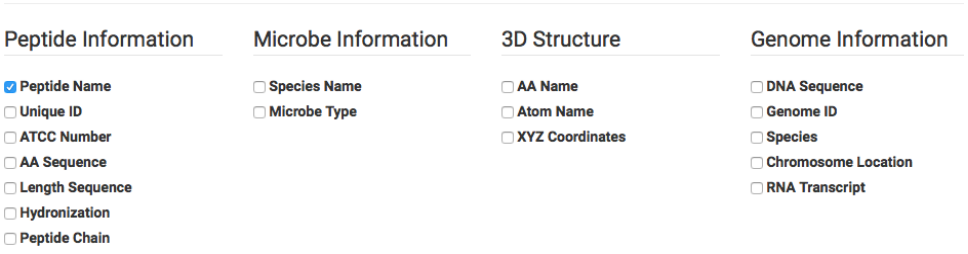


Figure 22: AMPed Summary Search Criteria Page

4.3.6. Summary Search Results

The summary search results page displays the relevant data to the user in an easy and intuitive way. It is a list of all the records that match a criteria that the user provideds. The results, as shown in the figure below, have a navigation link for each record that can be used to get further details. The results page displays the number of records found and automatically sorts the data on Unique ID.

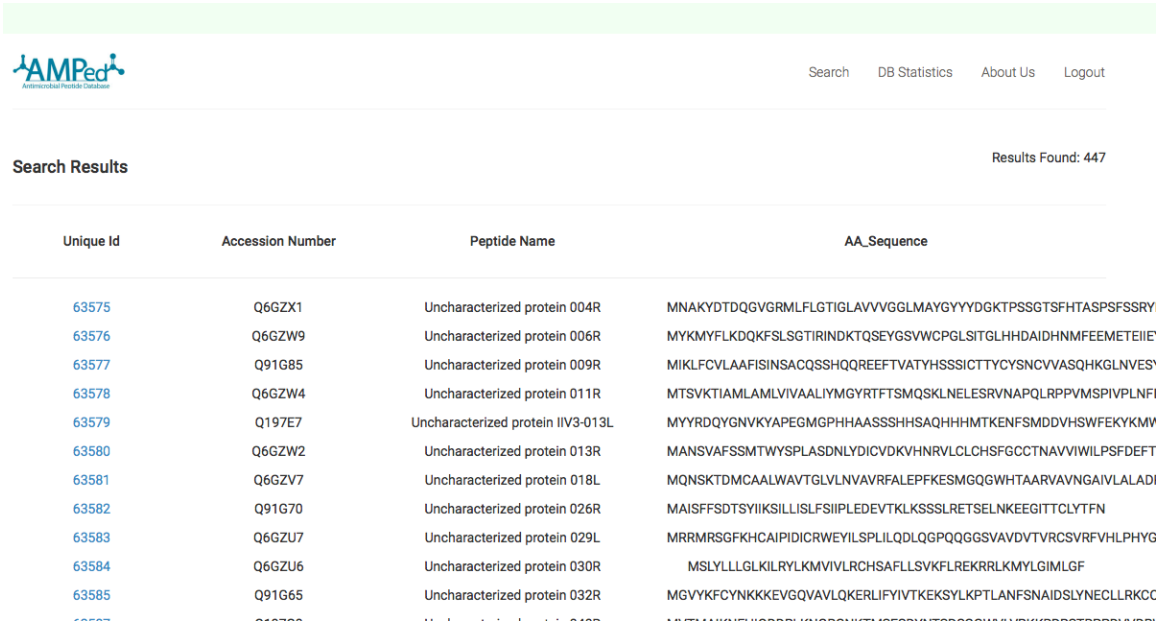


Figure 23: AMPed Summary Search Results Page

4.3.7. Detailed Search Results

A detailed search results page is navigated to when a user clicks for detailed data on the summary results page. The detailed search results page displays all the data available for a particular peptide in a single view.

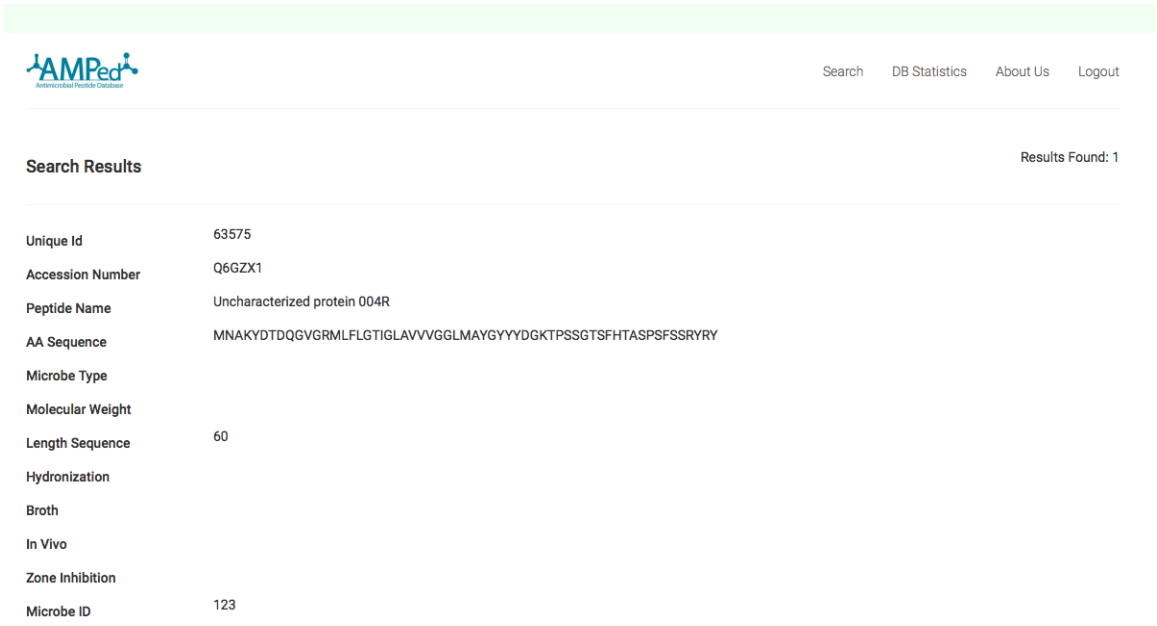


Figure 24: AMPed Search Detail Results Page

Search is one of the most critical user functionality in AMPed. In the next chapter, we will take a deeper look into how the AMPed search capability works.

4.4 Search

In this section of the paper, we will dive deeper into AMPed's search capability that makes heavy use of PHP and structured SQL queries to retrieve data and present relevant results to the researchers. Engineering the AMPed search functionality was a challenging task. The AMPed search needed to be flexible to meet the needs of the range from expert to novice users and be able to process a variety of criterions through tens of thousands of records in the AMPed database tables. The

search had to be designed for speed and accuracy. The search also had to manage a variety of data types.

Apart from applying the traditional database search query techniques, there was a technical challenge in designing the AMPed search due to the need to parse the very long text strings of Amino Acid (AA) Sequences to enable full and partial matches on the data. The full and partial search option on AA sequences allows researchers to quickly and accurately find results that might be of interest to them. The example below highlights how a full and partial string search using an amino acid sequence provides different results when searched via AMPed.

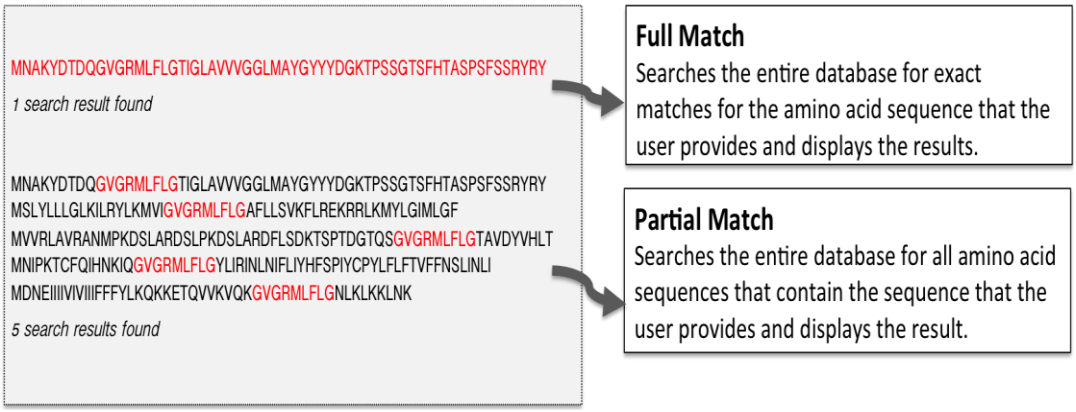


Figure 25: Types of Search implemented in AMPed

4.4.1. Search Design Goals

AMPed is used by a varied set of researchers whose time is very precious. It relies on a huge database with a many different data elements. Further, the data within AMPed is growing at a very rapid pace. Thus the AMPed search capability was designed with the following two primary goals:

- **Quality:** AMPed has a large amount of data and as research continues, it is expected to grow at a very rapid pace. Thus one of the main problems for AMPed search was that the number of documents will be increasing by many orders of magnitude, but the user's ability to look at documents will not. People will still only be able to consume only the first few tens of results. Because of this, as the collection size grows, we needed AMPed to have very high precision and return only the most relevant documents. Indeed, we wanted our notion of "relevant" to only include the very best documents since there may be tens of thousands of slightly relevant documents. This very high precision but is important for AMPed search even at the expense of recall i.e. the total number of documents the system is able to return.
- **Speed & Scale:** Aside from quality, AMPed had to be designed for speed and scale. AMPed had to be designed to deal with the ever growing data and meet the constant user expectation of getting results in a few seconds to minutes. In implementing AMPed, we saw bottlenecks in CPU, memory access, memory capacity, disk seeks, disk throughput, disk capacity, and network IO. AMPed had to be developed to overcome a number of these bottlenecks. We used the agile development process to develop modular SQL queries that can easily include new data elements and an extensible web page template for both - search criterion selection and results display page. Further, the results pages were broken into summary and detailed results to limit the data processing needs and provide results faster.

4.4.2. Search Queries

SQL (Structured Query Language) is a special-purpose programming language designed for managing and accessing data held in a relational database management system (RDBMS). AMPed data, as described earlier, is stored in a RDBMS. AMPed search primarily uses SQL queries, to process and retrieve the results from the AMPed database. Now, let's take a closer look at a few AMPed search queries. [2]

4.4.3. Fetching Data: SQL SELECT Queries

As part of this thesis, we spent a considerable amount of time developing means to fetch and display peptide data. The main challenge in designing the AMPed search feature was that the user wanted to "slice and dice" data in varied ways. That is, they wanted to look at the data and analyze it in an endless number of different ways, constantly varying the filtering, sorting, and calculation rules on the raw data. AMPed used the SQL **SELECT** statement to choose the relevant data that users wanted returned from the database [3][4][2]. The following example shows how AMPed used the SELECT statement to retrieve data that matched a particular peptide name:

First, connect to the database:

```
//Connect to the AMPed Database;
$hostname='***.*.*.*';
$usrnm='*****';
$pwd='**********';
$dbname='AMPed';
$con = mysql_connect($hostname, $usrnm, $pwd) OR DIE (mysql_error());
mysql_select_db($dbname);
```

Then, run the SQL query to fetch the relevant data.

```
//Select data when Peptide Name is entered only;
$sqlp = "Select Unique_ID, Accession_No, Name, AA_Sequence from Peptide
where Name Like '%$PepName%'";
$resultp = mysql_query($sqlp, $con) or die(mysql_error());
```

Select the tubles from the peptide table that have a particular peptide name (entered by the user) and projecting out only the Unque_ID, Accession_No, Name, AA_Sequence.

4.5. Displaying Data: Summary & Detailed Results PHP

Once the data is retrieved using the SQL SELECT statements, AMPed used **PHP** to massage and then display that data into the AMPed web interface. AMPed uses a three stage approach to compute and display data:

- **Stage 1:** Include the whole content retrieved by the SELECT query into a PHP array.
- **Stage 2:** Compute and then count the number of rows of data.
- **Stage 3:** Set up a loop that will take each row of the result and echo (i.e. display) the relevant data on AMPed web interface.

The following example shows how AMPed used this three stage process to display summary results.

```
//Stage 1: include content in a PHP array;
        $resultp = mysql_query($sqlp, $con) or die(mysql_error());

//Stage 2: Count the number of rows of data;
        $rowp = mysql_num_rows($resultp);

//Stage 3: Set up a loop, Process and then display the data;
        if($rowp == 0)
                {
                 echo '<div class="container">
                <div class="row">
```

```
            <div class="col-sm-12">
             <h4 style="text-align: left;">Search Results</h4><hr><br>
            </div></div></div>';
                    echo "No Results Found.";
            }else
            {
            echo ' <div class="header-middle"><div class="container">
           <div class="row">
            <div class="col-sm-4">
             <h4 style="text-align: left;">Search Results</h4>
            </div>
            <div class="col-sm-8">
            <div class="shop-menu pull-right"> Results Found: '; echo $rowp; echo
'</div></div></div></div></div><br>';
            echo '<div class="container">
            <div class="row">
            <div class="col-sm-2"> <h5> Unique Id </h5> </div>
            <div class="col-sm-2"> <h5> Accession Number </h5> </div>
            <div class="col-sm-3"> <h5> Peptide Name </h5> </div>
            <div class="col-sm-5"> <h5> AA_Sequence </h5> </div></div><hr>';
            while($rows=mysql_fetch_assoc($resultp)){
        echo '<div class="row">';
        echo '<div class="col-sm-2">'; echo '<p> <a
href="detail_result.php?UQ_ID=';
        echo $rows['Unique_ID']; echo '">'; echo $rows['Unique_ID']; echo
'</a></p></div>';
        echo '<div class="col-sm-2">'; echo '<p>';
        echo $rows['Accession_No']; echo '</p>';echo '</div>';
        echo '<div class="col-sm-3">';echo '<p>';
        echo $rows['Name']; echo '</p>';echo '</div>';
        echo '<div class="col-sm-5">'; echo '<p>';
        echo $rows['AA_Sequence']; echo '</p>';echo '</div>';
                    echo '</div>';

        }
         echo '</div>';
            }
```

4.6. SECURE ACCESS AND AUDIT TRAIL

In this section of the paper, we will dive deeper into AMPed's security capability, specifically the log-in function that was designed to provide secure user access and maintain AMPed's Audit trail capability. Let's do a deep dive first into AMPed Log-in feature that can be, at a high level, broken down into two parts:

- **CAPTCHA:** Prevent automated programs from performing functions that are supposed to be performed by human visitors
- **Username and Password:** Validate log-in credentials to provide access to secure content

4.6.1. CAPTCHA

In today's age of spamming, website managers have adopted techniques to prevent automated programs from performing functions that are supposed to be performed by human visitors to the site, such as log-in. AMPed has achieved this through implementation of CAPTCHA. CAPTCHA is a program that can prevent a computer from looking like a human. It is a loosely contrived acronym meaning "completely automated public Turing tests to tell computers and humans apart." [5] CAPTCHAs are graphics presented with distorted text. In AMPed, it is placed within the login form. AMPed websites uses CAPTCHA to specifically prevent abuse from "bots," or automated spamming programs that might try to login to AMPed website. The computer programs can not read distorted text as well as humans can, so bots are

less likely to enter the AMPed site protected by CAPTCHA. The process involves generating letters and numbers that appear in a distorted graphic and a text entry field is provided for users to enter the text of the CAPTCHA image.

The following provides a CAPTCHA script that is designed to thwart bots from logging into the AMPed site. The script of CAPTCHA used is stored in file captcha-image.php. The figure below shows the image of CAPTCHA created for the AMPed login page.

Figure 26: Image of CAPTCHA created for the AMPed login

To create CAPTCHA for AMPed, this thesis used PHP's GD library image functions [24][6]. The GD Graphics Library is a graphics software library for dynamically manipulating images. GD can create images composed of lines, arcs, text, other images, and multiple colors. Its native programming language is ANSI C. It will create JPEG images for AMPed site. While GD an be used to creat image files in a variety of different image formats, including GIF, PNG, JPEG, WBMP, and XPM, for AMPed CAPTCHA, we chose and built the image in JPEG format.

It has been observed that users often confused with few letters and numbers when using features like CAPTCHA as they are hard to recognize. So, to ensure ease of use, we also configured CAPTCHA code to not use, letters - "O", "I", "l" and numbers - "0", "1". These can be confusing for a user to interpret in an image. This

was achieved by building a small configuration in the code so that it can be easily edited when needed.

Note: GD library is enabled by default as of PHP 4.3 and was an option before that.

The following describes the CAPTCHA set for the AMPed login page.

Start the New Session

```
session_start();
header ('Content-type: image/png');
```

Destroy the session if already there

```
if(isset($_SESSION['amped_captcha']))
{
unset($_SESSION['amped_captcha']);
}
```

Enetr Alphabets and Numbers to Display

```
$string1="abcdefghijkmnpqrstuvwxyzABCDEFGHIJKMNPQRSTUVXYZ";
$string2="23456789";
$string=$string1.$string2;
$string= str_shuffle($string);
$random_text= substr($string,0,4);
```

Randomly assign 4 alphabets or Numbers to Session

```
$_SESSION['amped_captcha'] =$random_text;
```

Create Image

```
$im = @ImageCreate (80, 30)
or die ("Cannot Initialize new GD image stream");
```

Set color of background of the image

```
$background_color = ImageColorAllocate ($im, 204, 204, 204);
```

Set color of the text to display in the image.

```
$text_color = ImageColorAllocate ($im, 51, 51, 255);
ImageString($im,5,5,2,$_SESSION['amped_captcha'],$text_color);
```

Memory allocation for the image

```
imagejpeg ($im);
```

Memory allocation for the image is removed

imagedestroy($im); .

4.6.2. Username and Password

Users are required to register as a member of AMPed. They are required to send their details to Professor Lenore M. Martin or the individual/team assigned this job. Their profile will be screened and checked before the administrator registers them as the member. AMPed is designed to have different levels of access. As part of user creation, administrator creates a user name, password and also puts the user into one of these access levels. This helps maintain appropriate controls for authorized access to read, write, edit, and append access.

The login validation is performed on the server side. Server page programs and algorithms are not visible to the clients, so it will be more secure and better approach to validate input on server side. For Amped, we used the server side script in PHP, MySQL and JavaScript to validate the input directly. As explained in the Log-in page section of this thesis, the user is required to enter the username, password and CAPTCHA letters to log-in. First CAPTCHA letters validation is performed before sending the request to the database for user name/password validation. This helps avoid the unnecessary interaction with the database and helps give more robust security. If CAPTCHA validation is passed, the entered username and password will be cross checked with the AMPed database. The user will be logged-in into the AMPed site only after successful validation of all the criteria.

4.6.3. Audit Trail (Maintaining user access logs)

We can get the Internet Protocol (IP) address of any visitor by using PHP. Finding the IP address is very important requirement where we store the users or visitors details. For security reasons we stores the IP address of our visitors who are doing any transaction online. This can be useful to track the visitors to AMPed site and help to analyze the type of visitors AMPed is receiving. In the case of malicious activity, the IP addresses can help understand the point of origin and to potentially block access. This ensures that the AMPed site continues to function normally. In AMPed, the IP address is fetched using the REMOTE_ADDR command in PHP. The variable logEntry contains all the information along with date and time stamp. [4] The data in logEntry is appended into the end of the text file amped_login_log. The sample below shows a few entries into the "amped_login_log":

```
::1,08-29-15 00:47:48:000000,,Captcha Failed
::1,08-29-15 00:55:47:000000,,Captcha Failed
::1,09-02-15 09:46:56:000000,test,Login Success
::1,10-03-15 12:52:32:000000,test,Login Failed
::1,10-03-15 12:52:52:000000,test,Login Success
::1,11-10-15 14:53:17:000000,user1,Captcha Failed
::1,11-10-15 14:53:27:000000,user1,Captcha Failed
::1,11-10-15 14:53:42:000000,user1,Captcha Failed
::1,11-10-15 14:53:57:000000,user1,Login Failed
::1,11-10-15 14:56:07:000000,user1,Captcha Failed
::1,11-10-15 14:56:34:000000,user1,Login Failed
::1,11-10-15 14:56:56:000000,user1,Captcha Failed
::1,11-12-15 07:11:22:000000,user1,Login Failed
::1,11-12-15 07:12:46:000000,user1,Login Success
::1,11-19-15 18:17:18:000000,user1,Login Success
127.0.0.1,11-22-15 17:57:19:000000,user1,Login Success
```

Figure 27: AMPed Log File

As can be seen, it captures the following information about the visitor:

- Date in format of mm-dd-yy
- Time in format of US/Eastern - Hours: minutes: seconds: milliseconds
- IP Address of the visitor
- User Name of the visitor
- Login Status – Login attempt is Successful or Failed

The script below shows how AMPed logs visitor details:

Captures the IP Address

```
$ip=$_SERVER['REMOTE_ADDR'];
echo "IP address="; echo $ip; echo "<br>";
```

Captures the Date and Time to visit AMPed website

```
date_default_timezone_set('US/Eastern');
echo date('m-d-y H:i:s:u'); echo "<br>";
```

Write the Information in the amped_login_log,txt file.

```
$logEntry= $ip . "," . date('m-d-y H:i:s:u') . "," . $_POST['UserName']

file_put_contents('amped_login_log.txt', $logEntry.PHP_EOL , FILE_APPEND);
```

REFERENCES

[1] Microsoft Corporation, "*Performance Testing Guidance for Web Applications*", Publisher: Microsoft Press, November 15, 2007.

[2] URL: www.mysql.com, Last accessed: 11/5/2015

[3] Janet Valade, "*PHP and MySQL For Dummies*", 4th Edition

[4] URL: www.php.net, Last accessed: 11/5/2015

[5] Makarov Alexander, "*Yii Application Development Cookbook",* Second Edition, Packt Publishing, 2013.

CHAPTER 5

CONCLUSION AND FUTURE WORK

This thesis work lays the foundation for the Anti-Microbial Peptide Editable Database ("AMPed") tool that enables researchers to efficiently search and view relevant data about antimicrobial peptides. It has developed:

- An extensible 3NF database
- A template driven web interface
- A robust search capability; and
- A secure login access function

The thesis also introduced a new Web Interaction Diagram technique that helps to capture information architecture and interaction design of a website in a single visual artifact.

In the future, as new peptide research continues, there may be a need to capture new data elements or relationship between them into the AMPed database. This may require further edits and enhancements to the database especially if new information needs to be stored or searched for. Also, researchers may want to manipulate or change existing data for example, write their own notes or link their data to AMPed data. This might require a new GUI. AMPed's template driven web interface designed as part of this thesis can be used for this purpose. Developers will not need to design from scratch, instead they can pick up an existing AMPed web page template, include

in it already designed components like a header or footer, leverage predefined styles and DIV tags for content (refer Chapter 6) to design and layout the new web page.

Previously an innovative program called “Bioparser”, developed by George Konstantinidis, was used to automatically extract and annotate useful data from various online databases and repositories. The program was primarily used to populate data in AMPed. Since, the AMPed database now has a new design and schema, the Bioparser will need to be edited and enhanced to meet the needs of the AMPed tool.

Security is also an evolving and ever growing field. As the usage of AMPed and its data grows, so will be the interest of hackers to harm the system. So in future, more work may need to be done to protect the AMPed site from vulnerabilities.

APPENDICES

Sample of the excel sheet compiled the agile stories used for the AMPed project.

Story Id	Epic	As a/An	I want to	So that	Notes	Priority	Status
1	Login	User	securely log in	I get access to AMPed update and maintenance features		H	In Progess
1.1	Login	Administrator	securely store user id & password in database	authorized users can login	Create table user in AMPed, create sample data for us	H	Complete
1.3	Login	User	enter my userid	I can loginto the system		H	Complete
1.4	Login	User	enter my password	I can log into the system		H	Complete
1.5	Login	System	validate login credentials	only authorized users access the database		H	Complete
1.6	Login	System	maintain the login session	I do not have to login again and again		H	
1.7	Login	System	timeout the session after certain time	unauthorized people do not gain access to the system		H	In Progess
1.8	Captcha	User	securely signin	unauthorized people or robots do not gain access to the system		H	Complete
1.9	Captcha	Administrator	create images at run time and pass them securely into the HTML headers	create captcha image through GD			Complete
1.11	Captcha	Administrator	include captcha in the login bricklet	it is contextually available at login and prevents denial of service attacks		H	In Progess
1.12	Logout	User	logout	unauthorized people do not gain access to the system		H	
2	Audit Trail	Administrator	know about the login events	I can get insights into any unauthorized activity		H	
2.1	Audit Trail	User	log user details	I get record of users to access AMPed		H	In Progess
2.2	Audit Trail	System	capture the date/time of last login	I have the audit trail		H	In Progess
2.3	Audit Trail	User	see the date & time of my last login	to ensure my account did not have unauthorized accessed		M	In Progess
3	Home Page	User	easily navigate AMPed website	I can search the APMed database			Complete
3.1	Home Page	Administrator	easily add additional pages (HTML Template)	maintaining website is easy	Create a HTML template for the website pages		Complete
3.2	Home Page	Administrator	easily apply style, color, fonts etc. across the website (CSS)	maintaining website is easy	Create master CSS stylesheets		Complete
3.3	Home Page	User	read about the features of AMPed	I can understand about the utility of the website	draft and finalise the content		Complete
3.4	Home Page	User	visually browse throught the website	I don't have to spend lot of time reading content	create images		Complete
4	About us	User	look about the AMPed team, location, research interest and how to becor	I understand and know about the AMPed research			In Progess
4.1	About us	User	know about the team	I can view the people behind AMPed research	content, images and layout		In Progess
4.2	About us	User	know about the location	I can view where AMPed is located	content, images and layout		In Progess
4.3	About Us	User	know about the research interest	I can get insights about the AMPed research	content, images and layout		In Progess
4.4	About Us	User	know how to join/become user of AMPed	I can have authorization to download/edit data	content, images and layout		In Progess
5	Search	User	search AMPed database	I can get desired results to help in my research		H	In Progess
5.1	Search	Administrator	connect search webpage with the AMPed database	user can retrieve search results	POC for Peptide Name(excat match)	H	Complete
5.2	Search	User	easily select the search criteria (POC)	get relevant records of peptides	content, images and layout		Complete
5.3	Search	User	easily select the search criteria (Expand on POC - Phase 1)	get relevant records of peptides	content, images and layout		
\	Search	User	easily select the search criteria (Expand on POC - Phase 2)	get relevant records of peptides	content, images, layout and scripts (if needed)		
5.5		User	search for an exact peptide sequence	I can get desired results to help in my research			In Progess
5.6	Search	User	search for partial peptide sequence	I can get desired results to help in my research		H	In Progess
5.7	Search	User	search for like peptide name	I can get desired results to help in my research		H	Complete
5.8	Search	User	search for like ATCC Number	I can get desired results to help in my research		H	
5.9	Search	User	search for like Species Name	I can get desired results to help in my research		H	
5.11	Search	User	search for like Unique ID	I can get desired results to help in my research		H	
5.12	Search	User	TBD search criteria	I can get desired results to help in my research			

Figure 28: Sample of Agile Stories used for AMPed project

List of all the tables with attribute, data type and details of the AMPed Database is given below.

Table: Peptide

Attribute	Data type	Detail
AMP_ID	varchar (10)	Unique identification number assigned to each antimicrobial peptide record stored in the AMPed. The format of its serial number is AMPXXXXX, X is the number.

Accession_No	varchar	Accession_No is obtained from ATCC, online biological culture repository. It's a unique key identification of each protein record in the ATCC. If the ATCC number is missing, then Unique_Id is assigned to the Accession_No of the same record.
Name	text	Name of the antimicrobial peptide
AA_sequence	varchar	The amino acid sequence of the peptide
Mol_weight	text	Molecular weight of the peptide sequence
Bioparser	int	Indicates whether the tool Bioparser uploaded the record or not. It is assigned a binary number. 1 means yes uploaded by the bioparser and 0 means not by bioparser.
Notes	text	Any comment, remark or note for the peptide record.

Length_seq	int	Total length of the amino acid sequence
Approval_Status	varchar	Shows status of the verification of information added/edited by the user of AMPed.

Table: Fight_Against

Attribute	Data type	Detail
AMP_ID	varchar	Unique identification number assigned to each antimicrobial peptide record stored in the AMPed.
Microbe_ID	int	Identification number of each microbe entry. The format of its serial number is MX, X is the number.

Table: Microbe

Attribute	Data type	Detail
Microbe_ID	int	Identification number of each microbe entry.

Species Name	text	Technical term/name used for a species in binomial nomenclature
Microbe_Type	varchar	Types of Microbes. The microorganisms or microbes that can cause disease come in different forms like virus, bacteria, fungi, protozoa, helminthes.

Table: Test

Attribute	Data type	Detail
MIC	float	Minimal Inhibitory Concentration
UOM	varchar	Unit of measurement for MIC
Microbe_ID	int	Identification number of each microbe entry.
Method_ID	varchar	Unique identification number assigned to each method used for experiment on peptides.
Test_ID	Varchar	Unique identification number assigned to each test entry.

Table: Method

Attribute	Data type	Detail
Method_ID	varchar	Unique identification number assigned to each method used for experiment on peptides. The format of its serial number is MDX, X is the number.
Method Name	varchar	Name of the method like In Vivo, Broth, Zone Inhibition
Description	text	Description of the method

Table: 3D_Structure

Attribute	Data type	Detail
AMP_ID	varchar	Unique identification number assigned to each antimicrobial peptide record stored in the AMPed.
PDB_ID	varchar	Unique accession or identification code for every molecular model in the Protein Data Bank (PDB).
Source_ID	varchar	Unique identification number

		assigned to each source used to scan structure of peptides. The format of its serial number is SX, X is the number.

Table: Amino_Acid_Address

Attribute	Data type	Detail
AMP_ID	varchar	Unique identification number assigned to each antimicrobial peptide record stored in the AMPed.
AA_Names	char	Name of amino acid
Chain	varchar	Chain name of a series of amino acids joined by peptide bonds
Length_seq	int	The length of the peptide sequence
Sequence _AA_No	double	Sequence number of amino acid
hydro	text	Hydrophobicity
Phi	int	Phi angle
Psi	int	Psi angle

Table: Atom_coord_Source

Attribute	Data type	Detail
Source_ID	varchar	Unique identification number assigned to each source used for identify and calculate structure of peptides.
Source_Name		Name of the source like X-Ray
Description		Description of the source name.

Table: Atomic_Coordinates

Attribute	Data Type	Detail
AA_Names	varchar	Name of amino acid
Atom_num	Int	Atom number
Atom_Names	Varchar	Name of atom
X	Float (8,4)	X coordinate
Y	Float (8,4)	Y coordinate
Z	Float (8,4)	Z coordiante
charge	varchar	Charge carried by the peptide
Error	Float (8,4)	Temperature factor

Table: Gene

Attribute	Data type	Detail
Genome_ID	int	Unique identification number assigned to gene in NCBI.
Genome_Name	text	Genome Name
DNA_seq	text	DNA sequence
chromosome_num	varchar	Chromosome number
chrom_address_start	int	Chromosome location start number
chrom_address_end	int	Chromosome location end number
Species	text	Species name
RNA_Sequence	text	RNA Sequence
DNA_ID	varchar	Unique identification number assigned to each DNA_seq
RNA_ID	varchar	Unique identification number assigned to each RNA_Sequence
AMP_ID	varchar	Unique identification number assigned to each antimicrobial peptide record stored in the

		AMPed.

Table: Article

Attribute	Data Type	Description
AMP_ID	varchar	Unique identification number assigned to each antimicrobial peptide record stored in the AMPed.
Abstract	text	Abstract of the Article
Doi	varchar	Digital Object Identifier. unique number assigned by the publisher, identifies the journal and individual article.
Volume	varchar	Volume of the Article
Year	year	Article published year
Authors	varchar	Author/Authors of the article.
Article_Title	varchar	Title of the article
Journal_Article	varchar	Title of the journal contains the article
Start_Page	int	Start page number of the

		article
Start_End	int	End page number of the article

Table: User

Attribute	Data Type	Description
UserName	varchar	Uniquely identifies members of the AMPed.
Password	varchar	Secret string of characters that allows user access to the AMPed website.
Address	varchar	Address of the user or his/her affiliation like Street name, house number
Contact_No	varchar	Contact number of the user
Email	varchar	Email address of the user
Access_Level	int	Identification number which defines permissions granted to the user
First_Name	varchar	First name of the user
Last_Name	varchar	Last name of the user
Affiliation	varchar	Name of the Affiliation user associate

Note	varchar	Any comment, remark
Job_Title	varchar	Job title of the user
Country_ID	varchar	Country where user/affiliation located
State	varchar	State of the country
Zip_code	varchar	Zip code/Pin code

Table: Access_ Level

Attribute	Data Type	Description
Access_Level	int	Unique identification number assigned to each access level stored in the AMPed. The format is single digit number like 1, 2, 3.
Description	text	Description of the access level. Shows permissions given to the user of the AMPed like read, write or read. E.g. user with permission read only can't add/edit the AMPed data.

Table: Country

Attribute	Data Type	Description
Country_ID	varchar	Unique identification number assigned to each country stored in the AMPed.
Country	char	Name of the Country

Table: Inserted_By

Attribute	Data Type	Description
AMP_ID	varchar	Unique identification number assigned to each antimicrobial peptide record stored in the AMPed.
UserName	varchar	Uniquely identifies members of the AMPed.

Table: Results_of_Test

Attribute	Data Type	Description
Microbe_ID	varchar	Identification number of each microbe entry. The format of its serial number is MX, X is the number.
Test_ID	varchar	Unique identification number assigned to each test entry.

Table: Used_Method

Attribute	Data Type	Description
Method_ID	varchar	Unique identification number assigned to each method used for experiment on peptides. The format of its serial number is MDX, X is the number.
Test_ID	varchar	Unique identification number assigned to each test entry.

BIBLIOGRAPHY

ATCC (American Type Culture Collection) worldwide repository and distribution center for cultures of microorganisms, URL: http://www.atcc.org, Last accessed: 10/28/2015

Beck, Kent "*Embracing Change with Extreme Programming*". Computer 32 (10): 70–77., 1999, doi:10.1109/2.796139

Benbasat, Izak and Peter Todd, "*An Experimental Investigation of Interface Design Alternatives: Icon vs. Text and Direct Manipulation vs. Menu*", International Journal of Man-Machine Studies

[4] Butler Christopher, "*The Strategic Web Designer*", Publisher: HOW Books, August 22, 2012.

[5] Codd, E.F., "*Further Normalization of the Data Base Relational Model*". (Presented at Courant Computer Science Symposia Series 6, "Data Base Systems", New York City, May 24–25, 1971.) IBM Research Report RJ909 (August 31, 1971). Republished in Randall J. Rustin (ed.), "Data Base Systems: Courant Computer Science Symposia" Series 6. Prentice-Hall, 1972.

[6] Edward R. Tufte, "*Visual Explanations: Images and Quantities, Evidence and Narrative*", April 1998

[7] Helander, Martin, "*Handbook of Human-Computer Interaction*", 1988

[8] Janet Valade, "*PHP and MySQL For Dummies*", 4th Edition

[9] Jansen, B. J., "*The Graphical User Interface: An Introduction*", SIGCHI Bulletin 30(2), 22-26, 1998

[10] Jesse James Garrett, "*A visual vocabulary for describing information architecture and interaction design*", URL: http://jjg.net/ia/visvocab/, Last accessed: 11/2/2015

[11] Konstantinidis George, Thesis on the design and populating of the AMPed database "*Antimicrobial Peptide Editable Database*", 2015

[12] Makarov Alexander, "*Yii Application Development Cookbook",* Second Edition, Packt Publishing, 2013.

[13] Microsoft Corporation, "*Performance Testing Guidance for Web Applications*", Publisher: Microsoft Press, November 15, 2007.

[14] Mikael Olsson, "*PHP Quick Scripting Reference*", Publisher Apress, 2013

[15] Miller, George A, "*The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. Psychological Review*", Vol.101. No.2:343- 352

[16] NCBI (National Center for Biotechnology Information) provides access to biomedical and genomic information, URL: http://www.ncbi.nlm.nih.gov, Last accessed: 10/28/2015

[17] PDB (Protein Data Bank) is a crystallographic database for the three-

dimensional structural data of large biological molecules, such as proteins and nucleic acids, URL: http://www.rcsb.org, Last accessed: 10/2/2015

[18] Sarna, David E. and George J. Febish, "*What Makes a GUI Work*?" Vol. 4., (July 15 1994)

[19] UniProt (Universal Protein Database) is a central repository of information on proteins, URL: http://www.uniprot.org, Last accessed: 10/28/2015

[20] URL: www.toadworld.com, Last accessed: 03/20/2015

[21] URL: www.php.net, Last accessed: 11/5/2015

[22] URL: www.mysql.com, Last accessed: 11/5/2015

[23] URL: https://golang.org, Last accessed: 10/2/2014

[24] Wickens, Christopher D., "*Engineering Psychology and Human Performance*", 2nd edition, 1992

[25] Wikipedia is a free encyclopedia, URL: https://en.wikipedia.org/wiki/Database_normalization, Last accessed: 09/20/2015